



Institut für Fertigungstechnik



Entwicklung eines Demonstrators zur Anbindung von fehlersicheren speicherprogrammierbaren Steuerungen an ein Cloud-System

Masterarbeit
von
Johannes Krasa

Technische Universität Graz

Fakultät für Maschinenbau und Wirtschaftswissenschaften

Institut für Fertigungstechnik

Univ.-Prof. Dipl.-Ing. Dr.techn. Rudolf Pichler

Graz, im September 2018

In Kooperation mit:

Siemens AG Österreich

SIEMENS

Ingenuity for life

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am 10.09.2018

A handwritten signature in blue ink, appearing to read 'Johannes Jauer', written in a cursive style.

(Unterschrift)

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

10.09.2018
(date)

A handwritten signature in blue ink, appearing to read 'Johannes Jauer', written in a cursive style.

(signature)

Abstract

This thesis is about the planning, coding, installation and the build-up of a demonstrator, to show the connectivity of fail-safe programmable logic controllers with a cloud system to generate and analyze data. The project is realized in cooperation with the Institute of Production Engineering and the Digital Factory Division of the Siemens AG located in Graz.

As cloud system the open Internet of Things operating system named MindSphere is used. Within MindSphere customers can illustrate the collected data in apps. In this thesis different sources of data are used, on one side values from the integrated temperature sensors of fail-safe programmable logic controllers are obtained and on the other side data is generated with help of a robot by manipulating end-position sensors and an emergency stop button. To activate the sensors a collaborative robot with a gripper is used. The thus generated signals are processed in the fail-safe programmable logic controllers. Later on, the signals are modified to enable the transport with help of an internet connection and a gateway to the cloud.

The used gateway requests the data and sends it to MindSphere for the purpose of gaining information. With this specific information acquisition production can be made more efficient and future statements in the sense of predictive maintenance are possible.

Kurzfassung

Diese Arbeit handelt von der Planung, der Programmierung, der Installation und des Aufbaus eines Demonstrators zur Anbindung von fehlersicheren speicherprogrammierbaren Steuerungen an ein Cloud-System.

Ziel ist es, Daten zu generieren, welche in einer Steuerung so modifiziert werden, dass diese von einer Cloud Applikation verarbeitet werden können.

Als Basis für das Projekt dient die offene Internet of Things- und Cloud-Plattform mit dem Namen „MindSphere“.

Die Daten werden dabei von unterschiedlichen Sensoren bereitgestellt. Einerseits werden die internen Temperaturüberwachungssensoren der fehlersicheren speicherprogrammierbaren Steuerungen genutzt, andererseits generiert ein kollaborativer Roboter Daten, indem er sowohl Endlagen-Sensorik, als auch einen Not-Aus-Taster betätigt. Die so erzeugten Signale werden in den zuvor programmierten speicherprogrammierbaren Steuerungen verarbeitet und so modifiziert, dass ein Gateway diese Daten über eine bestehende Internetanbindung an die Cloud senden kann.

Mithilfe dieser Daten ist somit eine Produktionsüberwachung für den Kunden möglich. Zusätzlich lassen sich durch Datenanalysen Vorhersagen im Sinne der „Predictive Maintenance“ treffen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ausgangssituation.....	1
1.2	Aufgabenstellung	1
1.3	Aufbau der Arbeit	1
2	Theoretische Grundlagen der Arbeit	2
2.1	Speicherprogrammierbaren Steuerung	2
2.1.1	Aufbau.....	2
2.1.2	Funktionsweise und Programmierung.....	4
2.1.3	Fehlersichere SPS.....	5
2.2	Industrie 4.0	8
2.3	Cloud Computing	14
2.4	IoT das Internet of Things.....	17
2.4.1	Industrial Internet of Things.....	18
2.5	MindSphere.....	19
2.5.1	Aufbau von MindSphere	20
2.5.2	Datentransport	22
3	Praktische Umsetzung des Demonstrators	24
3.1	Aufbau der Roboterzelle und Sensorik zur Datengenerierung	24
3.1.1	Unterbau	24
3.1.2	Roboter	25
3.1.3	Demo-Cases und Sensorik.....	27
3.2	Aufbau der speicherprogrammierbaren Steuerung.....	33
3.2.1	Hardware	33
3.2.2	Software	38
3.3	MindSphere Einrichtung.....	57
3.3.1	Vorinstallationen am Computer	57
3.3.2	Hardware	58
3.3.3	IoT Data Modeler:	59
3.3.4	Apps.....	65
4	Ergebnis	77
4.1	Bewertung von SPS und MindSphere	77
4.1.1	SPS in Aufbau & Inbetriebnahme:.....	77

4.1.2	SPS im laufenden Betrieb:	77
4.1.3	MindSphere in Aufbau & Inbetriebnahme:	77
4.1.4	MindSphere im laufenden Betrieb:	78
4.2	Erweiterungen	79
4.3	Ausblick & kritische Betrachtung	79
	Literaturverzeichnis	81
	Abbildungsverzeichnis	83
	Tabellenverzeichnis	86
	Abkürzungsverzeichnis	87
	Anhang.....	88

1 Einleitung

1.1 Ausgangssituation

MindSphere ist eine cloudbasierte Software-Infrastruktur, die zum Sammeln und Analysieren von Prozessdaten – typischerweise aus Fertigungsprozessen - entwickelt wurde. Dieses System wird seit 2016 von der Digital Factory, einer Sparte der Siemens AG, entwickelt.

Das Siemens Development House Graz entwickelt Firmware für sichere speicherprogrammierbare Steuerungen. Diese Steuerungskomponenten werden auch Fail-Safe Module (FSM) genannt. In dieser Arbeit soll diese Kompetenz gezeigt werden, indem Daten mit Hilfe von FSM in MindSphere übertragen und dort gesammelt werden.

Die Ausgangssituation ist die, dass diese FSM noch nicht an MindSphere angebunden worden sind und bisher kein Datentransport möglich war.

1.2 Aufgabenstellung

In Kooperation mit der Siemens AG und der Smart Factory der TU Graz wurde in dieser Masterarbeit ein Demonstrator errichtet, der FSM an MindSphere anbindet, Daten generiert und diese in MindSphere auswertet und darstellt.

Dazu war es notwendig, die entsprechende Hardware, vom Grundgestell wie dem Tisch, über den Roboter, als auch die Hardware und Sensorik für die speicherprogrammierbare Steuerung auszuwählen. Nach dem Aufbau musste die Programmierung vorgenommen werden und das System so eingerichtet werden, dass spezifischen Daten aus den FSM in der Cloud verarbeitet werden können. Ziel der Arbeit war es, hierfür einen kontinuierlichen Datenfluss zu generieren. In weiterer Folge soll das System in den späteren Produktionsablauf der Smart Factory integriert werden können und auch als Digital Twin abgebildet werden. Mit Hilfe von MindSphere und den gesammelten Daten sollen Systeme effizienter gestaltet werden, Fehler vermieden oder bessere Vorhersagen über zukünftige Probleme gemacht werden können.

1.3 Aufbau der Arbeit

Zu Beginn der Arbeit werden Themen wie Industrie 4.0, Cloud-Computing und das Thema der speicherprogrammierbaren Steuerungen behandelt. Im nächsten Schritt wird der Aufbau der Hardware gezeigt. Danach wird die benötigte Programmierung und die Installation der MindSphere Verbindung beschrieben. Zum Abschluss wird die Funktionsfähigkeit beschrieben und eine Zukunftsperspektive dargestellt.

2 Theoretische Grundlagen der Arbeit

In den theoretischen Grundlagen der Arbeit werden die Themen Industrie 4.0, Cloud-Computing und Internet of Things behandelt, sowie ein Überblick über sichere speicherprogrammierbare Steuerungen gegeben.

2.1 Speicherprogrammierbaren Steuerung

Dieses Kapitel handelt vom Aufbau und der Funktionsweise von speicherprogrammierbaren Steuerungen (SPS) sowie von sicheren speicherprogrammierbaren Steuerungen.

Unter einer SPS versteht man Steuerungen auf digitaler Basis, die das Regeln, das Steuern und das Überwachen von Anlagen und Maschinen übernehmen [1].

2.1.1 Aufbau

Eine SPS ist in den meisten Fällen modular aufgebaut. Dabei übernehmen die verschiedenen Module jeweils eine eigene Aufgabe. In Abbildung 2.1 ist exemplarisch eine SPS dargestellt, deren wichtigste Komponenten den nächsten Punkten beschrieben werden [1]:

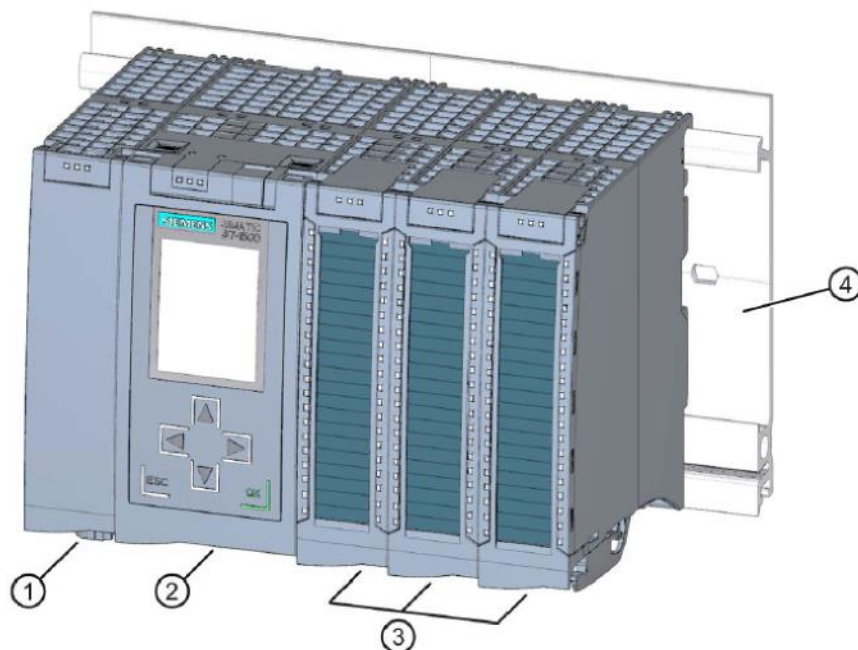


Abbildung 2.1: Speicherprogrammierbare Steuerung [2]

1. Spannungsversorgung:

Versorgt alle SPS Module mit Spannung.

2. CPU:

Die CPU (Central Processing Unit) ist der wichtigste Teil der SPS. Auf der CPU befindet sich ein Betriebssystem, welches das vom SPS-Programmierer entwickelte Programm ausführt. Dieses Programm wird auch Anwenderprogramm genannt und meistens mittels SD-Karte (Secure Digital Memory Card) auf die CPU übertragen. Zusätzlich zu den Stromanschlüssen besitzt die CPU einen LAN (Local Area Network) Anschluss. Dies ermöglicht, weitere Module wie z.B. dezentrale Peripherieeinheiten, Gateways für Cloud-Verbindungen oder den Rechner zum Programmieren anzuschließen.

3. Peripheriemodule:

Es gibt zentrale und dezentrale Peripheriemodule, welche sich wie folgt unterscheiden.

a) Dezentrale Module:

In Abbildung 2.2 ersichtlich sind dezentralen Module, wie sie in dieser Arbeit verwendet und auch bei Siemens Graz entwickelt werden. Der Unterschied besteht im Ort der Anbringung. Dezentrale Module sind direkt an Maschinen angebracht und werden mittels PROFINET (Process Field Network), einem LAN System, das Daten mittels Siemens-Protokoll überträgt, an die CPU angeschlossen. Daraus ergibt sich der Vorteil kurzer Verdrahtungen von der Sensorik zu den Modulen. Daten können somit an dezentralen Punkten gesammelt und einfach mittels LAN Kabel auf eine zentrale CPU übertragen werden.



Abbildung 2.2: SPS dezentraler Peripherie [3]

b) Zentrale Module:

Zentrale Module, wie in Abbildung 2.1 ersichtlich, sind direkt mittels Rückwandbus mit der CPU verbunden.

SPS Module werden je nach der Art der zu verarbeitenden Daten unterschieden. Hierbei gibt es mehrere Varianten [1]:

- **Digitaleingabebaugruppe (DI)**
Eine DI nimmt binäre Signale (1-Bit) von Schaltern oder Tastern auf.
- **Digitalausgabebaugruppe (DQ)**
Eine DQ schickt binäre Signale (1-Bit) an Schalter, Ventile etc.
- **Analogeingabebaugruppe (AI)**
Eine AI nimmt analoge Signale von Sensoren auf und wandelt diese zur Weiterverarbeitung in binäre Signale um.
- **Analogausgabebaugruppe (AQ)**
Eine AQ wandelt binäre Signale der Steuerung in analoge Signale um.

4. Montageschiene:

Die Montageschiene dient der standardisierten Positionierung und Fixierung der Module im Schaltschrank.

2.1.2 Funktionsweise und Programmierung

Das Anwenderprogramm wird mittels einer spezifischen Software auf einem PC programmiert. Für Siemens-Komponenten verwendet man dafür das Programm TIA Portal von Siemens. Die genaue Vorgehensweise sowie die Möglichkeiten werden in Kapitel 3.2.2 beschrieben.

Nach dem Übertragen des Anwenderprogramms auf die CPU lässt diese das Anwenderprogramm zyklisch ablaufen. Dazu werden im ersten Zustand die aktuellen Signalzustände eingelesen, danach wird das Anwenderprogramm aufgerufen und die Anweisungen der Reihe nach ausgeführt. Ist dieser Prozess abgeschlossen, werden die Änderungen an den Ausgängen ausgegeben und der Programmablauf beginnt von vorne. Zu beachten ist die Zykluszeit, also jene Zeit die benötigt wird, um das Anwenderprogramm einmal ablaufen zu lassen. Wichtig ist hierbei, dass Signaländerungen an den Eingängen

rechtzeitig erfasst und so die richtigen Ausgangssignale ausgegeben werden können. Ist das nicht der Fall, ist die Steuerung nicht echtzeitfähig [1].

Ein System ist echtzeitfähig, wenn es innerhalb einer festgelegten Zeitspanne reagiert. Diese Zeit kann beliebig und je nach Anforderung für jedes System gewählt werden. Moderne SPS arbeiten mit einer Zykluszeit von 5-30ms, die Zeiten können aber je nach Programmlänge und Anforderung variieren [4].

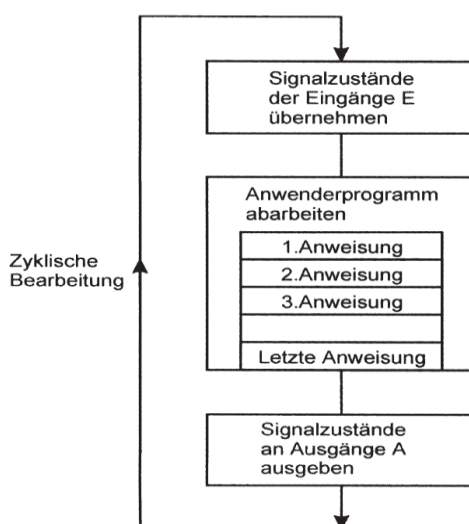


Abbildung 2.3: Ablauf der zyklischen Bearbeitung eines SPS Programms [1]

2.1.3 Fehlersichere SPS

Eine der Hauptkompetenzen der Digital Factory Division der Siemens AG ist die Entwicklung von Firmware für fehlersichere SPS Module, konkret auch F-Module, oder wie oben bereits erwähnt, Fail-Safe Module (FSM) genannt.

Diese besitzen im Gegensatz zu normalen Modulen Sicherheitsfunktionen, die sicherheitsrelevanten Abläufe steuern und regeln. Als sicherheitsrelevante Abläufe gelten Prozesse, die Menschen gefährden können. Genormt sind diese nach der DIN 61508, welche verschiedene Safety Integrity Level (SIL1 - SIL4) definiert [5]. Fail-Safe Module müssen einen fehlersicheren Betrieb garantieren. Im Falle eines Fehlers muss das Arbeitssystem in einen fehlersicheren Laufzustand gebracht werden und so einen sicheren Zustand einnehmen. Daher werden die meisten dieser Steuerungen redundant ausgeführt [1].

In der Praxis wird das meistens mit einem „1oo2“-System umgesetzt, wie es auch in dieser Arbeit bei der Verkabelung eines Not-Aus-Tasters verwendet wird. Unter „1oo2“ versteht man die Bezeichnung „one out of two“. Dieses System verhindert im Falle eines Fehlers einen gefährlichen Zustand [6]. Die Funktionalität wird anhand eines Beispiels erläutert.

Abbildung 2.4 zeigt einen Sensor, der nur nach der erfolgreichen Signalbearbeitung von zwei Rechnern ein Ausgangssignal zum Start eines Aktors sendet. Fällt z.B. Rechner 2 aus, ist

der Ausgang 2 nicht mehr in der Lage, die gesamte Versorgungsleitung zu schließen und somit kann der Aktuator nicht gestartet werden. Somit ist das „1oo2“ Prinzip erfüllt, sprich das System ist sicher und funktioniert eben nur, wenn beide Rechner korrekt arbeiten.

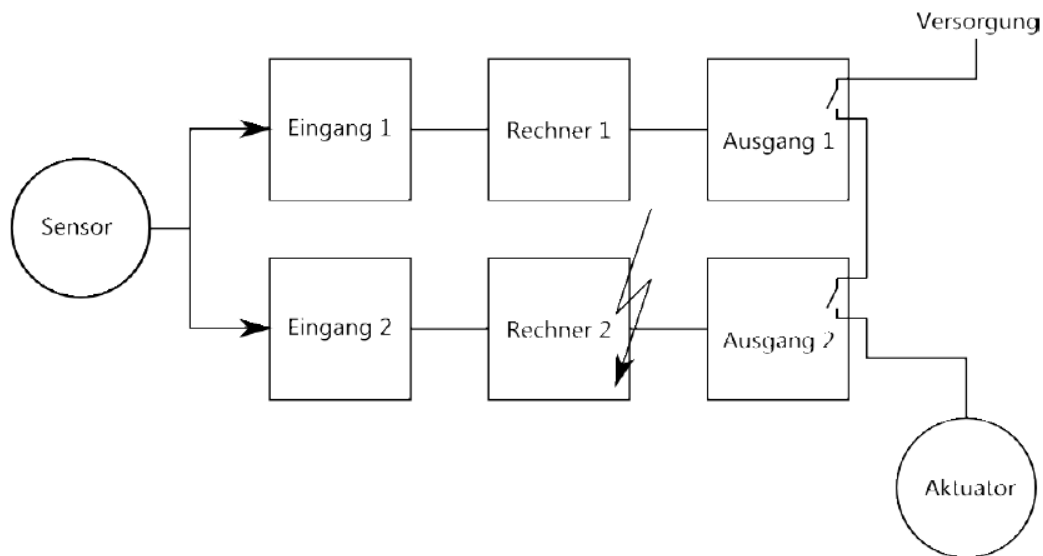


Abbildung 2.4: 1oo2 Schaltprinzip [6]

Im nächsten Punkt wird ein FSM beschrieben. Dabei gibt es verschiedene Modularten. Siemens entwickelt F-DI, F-AI, und F-DQ Module, also fehlersichere Digitaleingabebaugruppen, Digitalausgabebaugruppen sowie Analogausgabebaugruppen. In Abbildung 2.5 ist ein Blockschaltbild eines solchen Fail-Safe Moduls dargestellt, welches in den nächsten Punkten kurz erklärt wird [7].

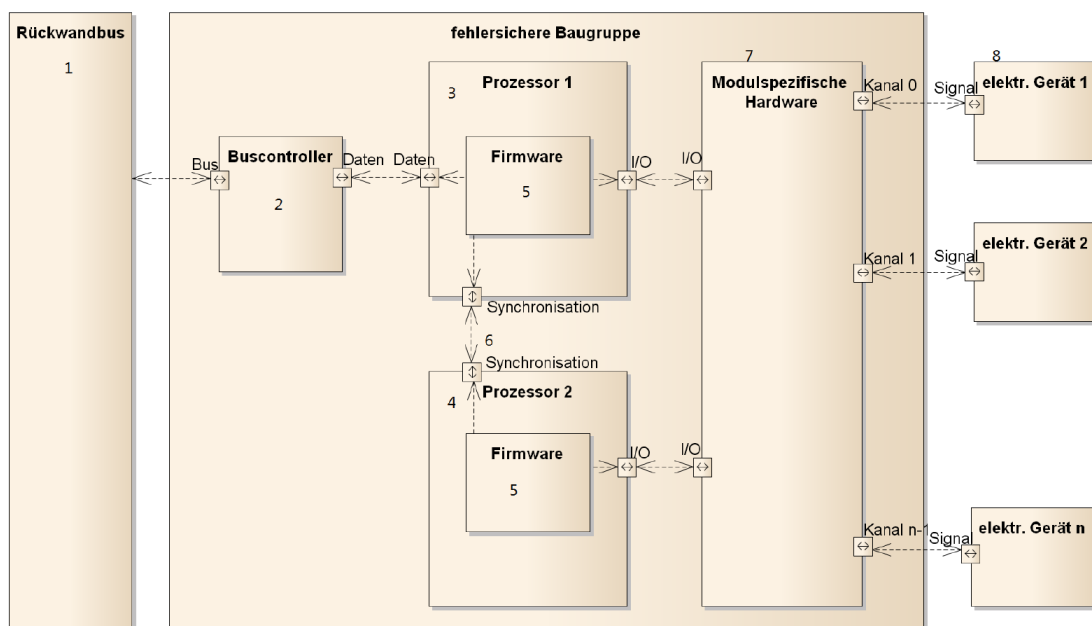


Abbildung 2.5: Blockschaltbild einer fehlersicheren Baugruppe [7]

1. **Rückwandbus:** Dient dem Datenaustausch der Module mit der CPU und befindet sich auf der Rückseite des Moduls
2. **Buscontroller:** Ist für den Datenaustausch zwischen dem Bus und den Prozessoren zuständig
- 3./4. **Prozessoren:** Auf den zwei identischen Prozessoren läuft die gleiche Firmware (5).
5. **Firmware:** Die Firmware ist eine im Gerät integrierte Software. Sie ist die Verbindungstelle zwischen der Anwendersoftware und der Hardware. Eine Firmware wird oft auch als Betriebssoftware von Geräten und Komponenten beschrieben [8].
6. **Synchronisationsschnittstelle:** Diese vergleicht und synchronisiert die Daten, um im Falle eines Unterschiedes (1oo2 Prinzip) Alarm zu schlagen.
7. **Modulspezifische Hardware:** Die Hardware unterscheidet sich dabei bei DI, DQ oder AI Modulen. Je nachdem welche Signale verarbeitet werden, müssen die Module in der Lage sein verschiedene Spannungen und Ströme aufzunehmen.
8. **Elektronische Geräte:** Dabei handelt es sich um Geräte, welche nach Bedarf angeschlossen werden können.

In Abbildung 2.6 ist die Platine eines F-DQ Moduls ersichtlich. Darauf zu erkennen sind zwei Prozessoren (CPUs) und der Anschluss für den Rückwandbus. Verwechselt werden dürfen diese CPUs nicht mit der Haupt-CPU der SPS.

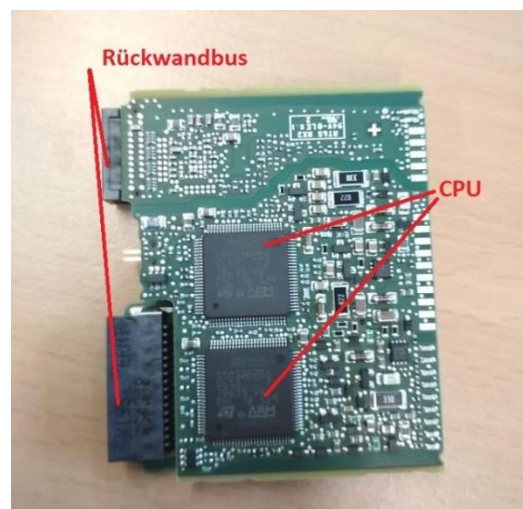


Abbildung 2.6: Platine einer F-DQ Baugruppe

2.2 Industrie 4.0

In Industrie 4.0 Architekturen spielen sicherheitsrelevante Anwendungen eine große Rolle. Trotz der fortschreitenden Digitalisierung sind Menschen im Produktionsablauf nach wie vor integriert. Um eine Produktion sicherheitstechnisch zu überwachen, können verschiedene Hilfsmittel, von Lichtschranken, Safety-Eyes, Sicherheitsmatten bis zu Not-Aus-Tastern verwendet werden. Um auch diese kritischen Einheiten überwachen zu können, werden Methoden aus der Industrie 4.0 angewendet. Damit der Begriff „Industrie 4.0“ von allen gleich verstanden wird, wird in dieser Arbeit zunächst ein kurzer Überblick gegeben. In den nächsten Punkten sowie in Abbildung 2.7 sind die 4 Stufen der Industriellen Revolution erklärt bzw. ersichtlich.

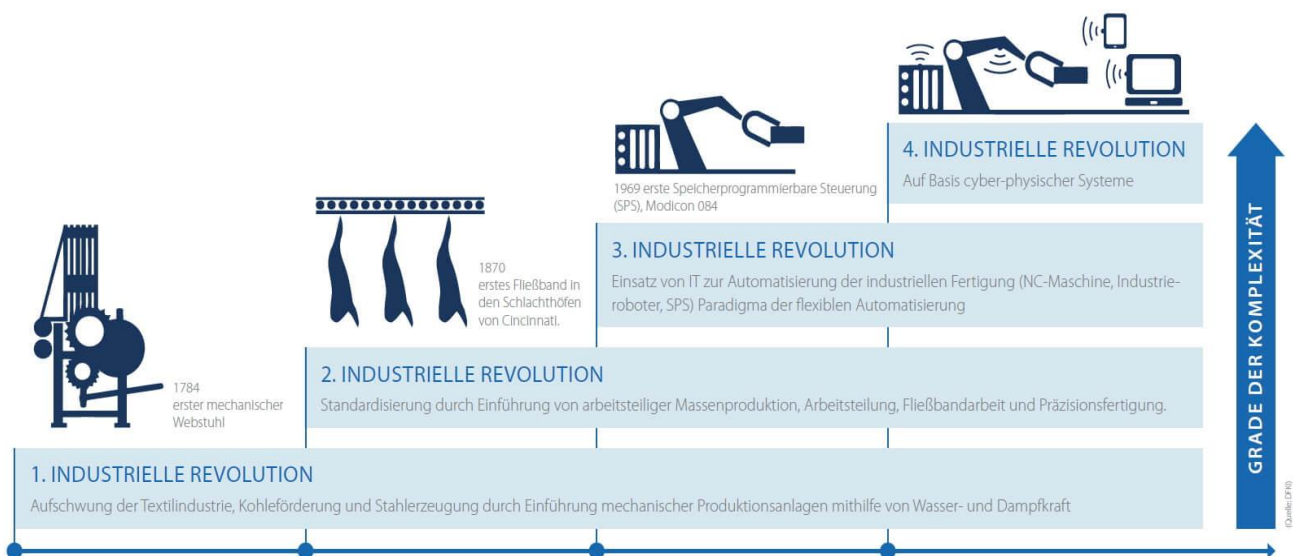


Abbildung 2.7: 4 Stufen der industriellen Revolution [9]

1. Die erste Revolution begann mit der Erfindung der Dampfmaschine im 18. Jhdt. (Abbildung 2.8). Durch die Weiterentwicklung und Anwendung dieser konnten weitere Erfindungen, wie der mechanische Webstuhl oder neue Verfahren in der Eisen- und Stahlerzeugung gemacht werden. Dies führte zu einem ersten großen Schub in der industriellen Produktivität [10].

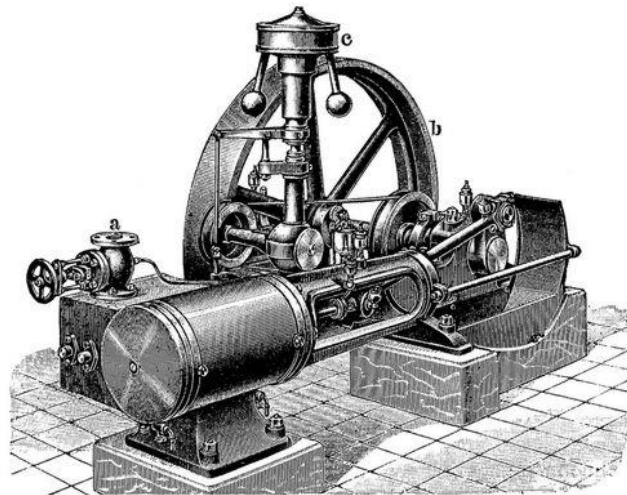


Abbildung 2.8: Dampfmaschine [11]

2. Zu Beginn des 20. Jhdt. begann der Einzug der Betriebswirtschaftslehre in die Fertigungsbetriebe. Diese erfolgte über die Aufteilung der Arbeitsschritte in kleinstmögliche Einheiten, um eine Produktion effizienter zu gestalten. Bekanntestes Beispiel ist hierfür die Produktion der Tin Lizzy von Henry Ford. Dieses Konzept der Produktion, auch Fließbandproduktion (Abbildung 2.9) genannt, verbreitete sich auf nahezu alle Industriesparten. Zusätzlich ermöglichte der Verbrennungskraftmotor und somit die Nutzung von Öl eine Beschleunigung der Entwicklung. Für die Gesellschaft bedeutete dies aufgrund leistbarer Produkte eine Erhöhung des Wohlstandes. Diese neuen Errungenschaften verursachten aber auch Spannungen, da aufgrund von Innovationen gewisse Berufe und Arbeiten verschwanden. Ähnlich wie es derzeit mit der Digitalisierung von Produktionen vorstatten geht [10].



Abbildung 2.9: Die erste Fließbandproduktion von Opel [12]

3. Zu Beginn der siebziger Jahre begann die dritte industrielle Revolution, in welcher Mechanik mit elektronischen Steuerungskomponenten das erste Mal vereinigt wurde (Abbildung 2.10). Als Beispiel sollte hier die SPS genannt werden, welche mit der Sensorik und der Aktorik die Grundlage jedes modernen Produktionssystems bildet. Außerdem legte die Mikroelektronik zu dieser Zeit den Grundstein für heutigen IT-Technologien [10].

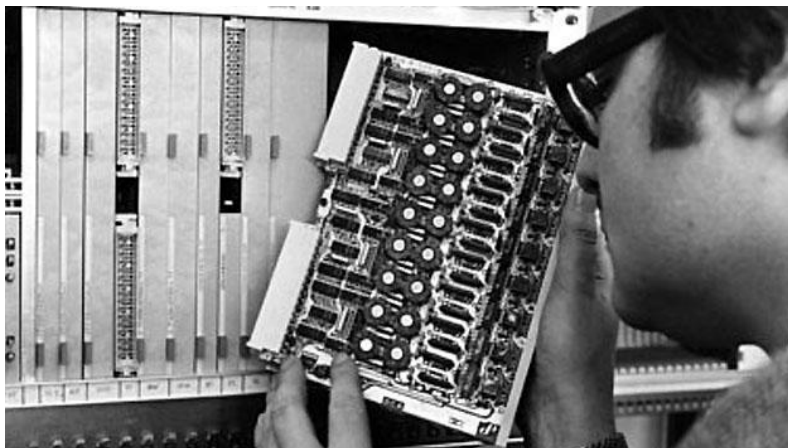


Abbildung 2.10: Erste Simatik SPS von Siemens [13]

4. Der Begriff Industrie 4.0 ist von der deutschen Bundesregierung geprägt worden, womit die durchgehende Digitalisierung der Produktion zu verstehen ist und der Name auch den Verweis auf die 4. Industrielle Revolution gibt [10]. Unter Digitalisierung versteht man die Erweiterung alter Technologien, wie zum Beispiel die Industrierobotik oder die Automatisierungstechnik, mit neuesten Informations-Technologien. Zu diesen neuen Technologien gehört Machine-Learning, Machine-Communication, Big Data Analytics, Virtual Reality, Cloud-Computing oder das Internet der Dinge [10, 14]. Systematisch sind die benötigten Kerntechnologien in Abbildung 2.11 dargestellt.



Abbildung 2.11: Kerntechnologien für die Industrie 4.0 [14]

Die 4. Revolution baut somit stark auf der dritten auf, verwendet deren Technologie, verbindet und verbessert diese. Mit Hilfe von Echtzeitdatenerfassung und der Möglichkeit Maschinen miteinander kommunizieren zu lassen, entsteht so aus einem Automatisierungssystem ein Konzept im Sinne der Industrie 4.0 [9, 10].

Die Industrie 4.0 beruht auf drei wesentliche Merkmalen, welche als gemeinsames Grundprinzip die Echtzeit besitzen (siehe Abbildung 2.12). Zusätzlich müssen diese Merkmale intelligent vernetzt werden.

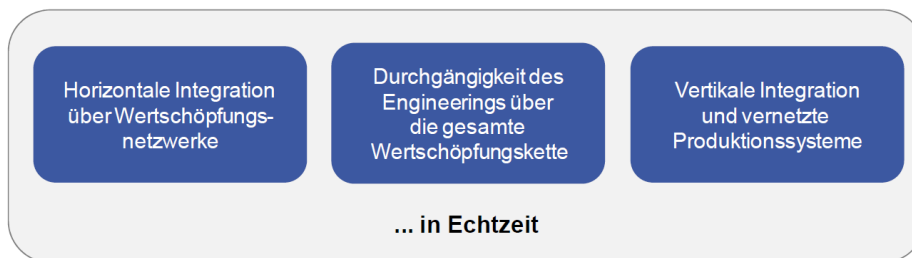


Abbildung 2.12: Merkmale der Industrie 4.0 [15]

- **Horizontale Integration über Wertschöpfungsnetzwerke**

Darunter versteht man die Vernetzung aller Prozessschritte in der Wertschöpfungskette. Dazu gehören die internen Bereiche, wie die Logistik, die Fertigung, der Vertrieb und andere nachfolgende Dienste, als auch externe Dienste, die von Partnern angeboten werden. Die Wertschöpfungskette wandelt sich so zu einem Wertschöpfungsnetzwerk [15] [16].

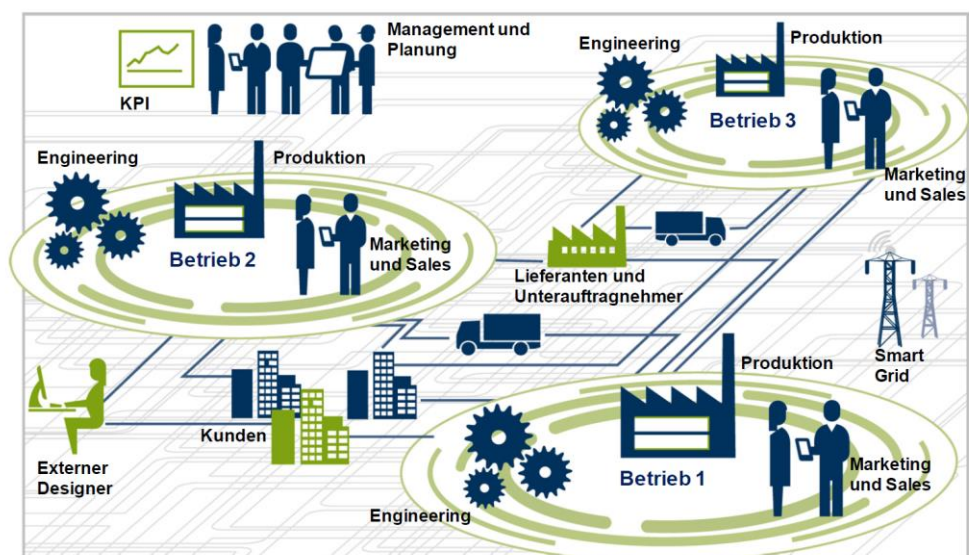


Abbildung 2.13: Wertschöpfungsnetzwerk [16]

- Durchgängigkeit des Engineerings über die gesamte Wertschöpfungskette**
 Ziel sollte es sein die reale Welt komplett virtuell abzubilden. In Zukunft soll es für den Kunden möglich sein, sein Produkt selbst zu konfigurieren und nicht aus dem Katalog eines Herstellers auswählen zu müssen. Mit Hilfe von digitalen Zwillingen sollte die gesamte Produkt-Architektur vom Engineering bis zur Herstellung abgebildet werden können [15] [16].

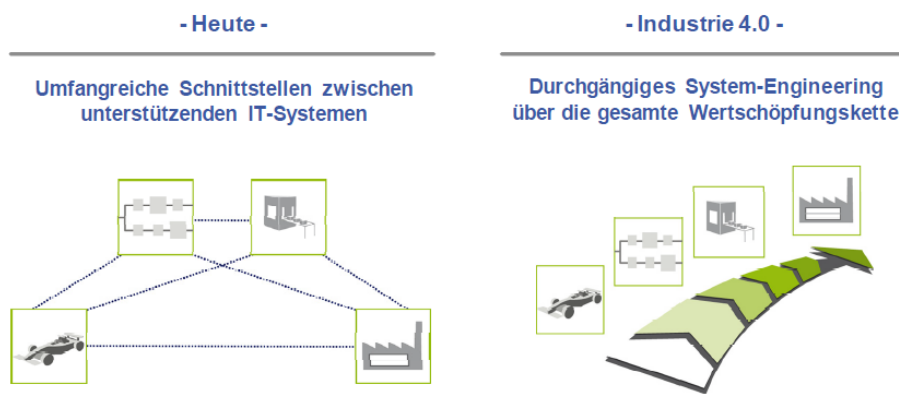


Abbildung 2.14: Durchgängiges System-Engineering [16]

- Vertikale Integration und vernetzte Produktionssysteme**

Unter diesem Merkmal ist die Verknüpfung und Vernetzung aller Hierarchieebenen, von der Feldebene bis zur Unternehmensführung zu verstehen. Es sollte dabei für jede Ebene die Möglichkeit bestehen, auf Daten aus einer anderen Ebene zuzugreifen. Dies funktioniert, wenn die unterschiedlichen Ebenen mit Hilfe von IT-Technologien zu einer Lösung verknüpft werden [15] [16].

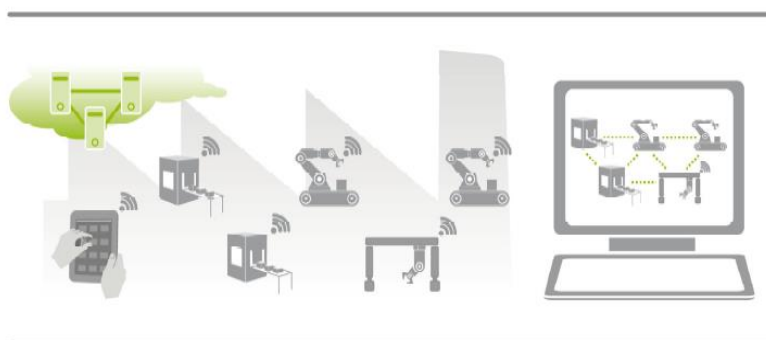


Abbildung 2.15: Vertikale Integration [16]

Im nächsten Punkt sollten die wichtigsten Potenziale von Industrie 4.0 und den disruptiven Technologien dargestellt werden [17]:

1. Individualisierung der Kundenwünsche:

Kunden können bei vertretbaren Preisen individuell ihr Produkt bestellen, Stichwort Losgröße 1.

2. Flexibilisierung:

Mit Industrie 4.0 wird die Fertigung flexibler. Der Anbieter kann schneller auf Stückzahl -oder Produktveränderungen reagieren und so effizient und kostengünstig produzieren.

3. Optimierte Entscheidungsfindung:

Durch Datenanalysen und echtzeitfähige Produktionssysteme werden, Entscheidungen, die die Produktion beeinflussen, leichter gemacht werden.

4. Ressourceneffizienz:

Auf Grund verbesserter und aktueller Informationen kann die Ressourceneffizienz gesteigert werden. Auch hier können mit Hilfe disruptiver Technologien Verbesserungspotenziale erkannt und behoben werden.

5. Wertschöpfungspotenziale durch neue Dienstleistungen und Fertigungstechniken:

Durch neue Technologien ergeben sich neue Wertschöpfungspotenziale und Dienstleistungen. Diese könnten Datenanalysen, Marktanalysen aber auch Fertigungstechniken wie der 3D Druck sein.

6. Wettbewerbsfähigkeit als Hochlohnstandort:

Mit Hilfe der Industrie 4.0 können Hochlohnländer ihre Standorte sichern, indem sie über den Einsatz dieser neuen Technologien neue Geschäftsmodelle generieren und neue Märkte erschließen sowie ihre Leistungsfähigkeit gegenüber dem Wettbewerb steigern

2.3 Cloud Computing

Ein wichtiger Bestandteil des Demonstrators ist die Datenverarbeitung in der Cloud (Cloud-Computing). Für Cloud-Computing gibt es verschiedene Definitionen, die bekannteste ist die Definition der US-amerikanischen Standardisierungsstelle NIST (National Institute of Standards and Technology) [18] [19]:

"Cloud Computing ist ein Modell, das es erlaubt, bei Bedarf, jederzeit und überall bequem über ein Netz auf einen geteilten Pool von konfigurierbaren Rechnerressourcen (z. B. Netze, Server, Speichersysteme, Anwendungen und Dienste) zuzugreifen, die schnell und mit minimalem Managementaufwand oder geringer Serviceprovider-Interaktion zur Verfügung gestellt werden können."

Fünf Eigenschaften muss ein System erfüllen um als Cloud-Computing zu gelten [18] [19]:

1. On-demand Self Service:

Automatische Bereitstellung der benötigten Ressourcen, wie zum Beispiel Rechenleistung, Speicher, Software usw.

2. Broad Network Access:

Der verwendete Service ist für das ganze Netz verfügbar, sprich überall erreichbar.

3. Resource Pooling:

Die angebotenen Ressourcen des Anbieters liegen in einem Pool vor und werden von mehreren Anwendern genutzt. Diese wissen in aller Regel nicht, wo sich die Ressourcen befinden.

4. Rapid Elasticity:

Das Service ist elastisch. Darunter versteht man die beliebige Skalierbarkeit der benötigten Services. Der Nutzer bekommt den Eindruck eines hoch flexiblen Systems, welchem unbegrenzte Ressourcen zur Verfügung stehen.

5. Measured Services:

Die Nutzung der Services und Ressourcen kann überwacht und gemessen werden. So lässt sich ein exakt passendes Modell für den Nutzer bereitstellen. Der Kunde muss, daher auch nur für die von ihm genutzte Leistungen bezahlen. Darunter versteht man auch das „Pay per Use“-Modell.

Unterscheiden lassen sich Cloud-Lösungen in den Bereitstellungsmodellen sowie in den Servicemodellen.

Unter Bereitstellungsmodellen versteht man die Zugänglichkeit für Nutzer. Es gibt folgende Modelle: die Private Cloud, die Public Cloud, die Community Cloud und die Hybrid Cloud.

Bei der Private Cloud, wird eine Cloud Infrastruktur nur für eine Institution verwendet. Bei der Public Cloud verwenden mehrere Anwender mit unterschiedlichem Interesse die gleiche Infrastruktur. Bei der Community Cloud teilen sich mehrere Anwender mit gleichen Interessen die Infrastruktur, die selber bereitgestellt wird. Unter Hybrid Cloud versteht man die gemeinsame Nutzung mehrere Cloud-Infrastrukturen, die trotzdem für sich eigenständig sind aber miteinander kommunizieren können [18] [19].

Die Unterscheidung der Servicemodelle kann wie folgt vorgenommen werden [18] [19]:

1. **Infrastructure as a Service (IaaS)**

Darunter versteht man die Bereitstellung von Infrastruktur wie Speicherplatz, Rechenleistung, Netzwerken und Kommunikationsgeräten wie Router, Switches oder Firewalls. Der Kunde baut sich seinen eigenen Service auf und nutzt nur die vorhandene Hardware und Infrastruktur.

Beispiel: Siemens MindSphere nutzt die SAP Cloud als IaaS in dem sie deren Hardware benutzen, aber ihre eigene Software verwenden. (Kunde ist Siemens, nicht der MindSphere User)

2. **Platform as a Service (PaaS)**

Der Kunde bekommt zusätzlich zur Hardware (IaaS) eine Plattform, auf der er selbstständig Software entwickeln und diese anbieten kann. Als Plattform kann man zum Beispiel ein Betriebssystem ansehen. Der Kunde von PaaS hat keine Möglichkeit auf das Betriebssystem oder die Hardware zuzugreifen.

Beispiel: Ein Hersteller verwendet Siemens MindSphere zur Prozessüberwachung. Auf die MindSphere spielt er dabei keine von Siemens zu Verfügung gestellte App, sondern seine eigene. Somit nutzt dieser Kunde MindSphere als PaaS, um seine Anwendungen auf MindSphere laufen zu lassen.

3. **Software as a Service (SaaS)**

Im letzten dieser drei Services erhält der Nutzer nur mehr die Möglichkeit, die auf der Cloud befindliche Software zu nutzen. Um wieder ein Beispiel zu geben, kann man sich einen Nutzer von MindSphere, der nur bereitgestellte Apps nutzt, vorstellen. Google Drive ist auch ein SaaS. Benutzer können Daten verarbeiten, Dokumente mit den vorhandenen Apps öffnen und bearbeiten, jedoch keine zusätzlich Anwendungen und Programme selbstständig hinzufügen und so die Funktion erweitern.

Um dies bildlich darzustellen, sind die einzelnen Verantwortungsbereiche in Abbildung 2.16 ersichtlich. OS steht dabei für Operating System (engl. für Betriebssystem). Unter Private Cloud ist das Erstellen eines eigenen Cloud Systems gemeint. Orange gekennzeichnete

Felder liegen dabei im eigenen Verantwortungsbereich, blaue Felder hingegen kennzeichnen Dienste, die als Service angeboten werden.

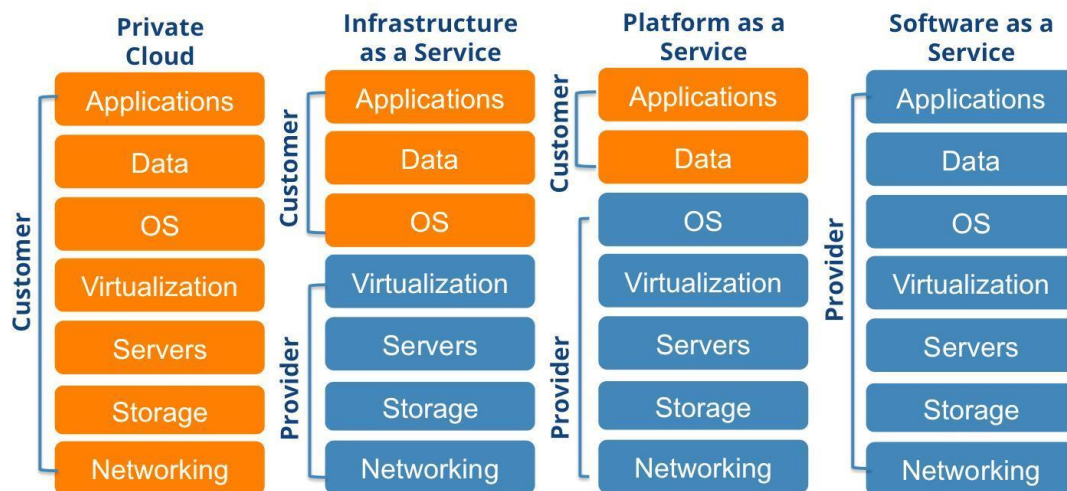


Abbildung 2.16: Darstellung der verschiedenen Servicestrukturen [20]

2.4 IoT das Internet of Things

Unter dem Begriff Internet of Things (IoT), auch Internet der Dinge, versteht man die Vernetzung von Gegenständen mit dem Internet, um deren Kommunikation zu ermöglichen. Dabei sollen diese Things mit Hilfe der Kommunikation selbstständig Aufgaben zugeteilt bekommen und diese verarbeiten. Bei diesen Things kann es sich um Sensoren, Maschinen, Anlagen oder um andere physische kommunikative Systeme handeln [21] [22].

Ziel ist es, ein Produktionsnetz zu schaffen, welches sich mithilfe der Fähigkeit zur Kommunikation sich selbst kontrolliert. Jeder Teilnehmer braucht dazu eine eindeutige Identifikation [22].

Siemens möchte mit MindSphere genau diese Kommunikation fördern, diese Things verbinden und so Systeme effizienter machen. Um das Potential von MindSphere zu veranschaulichen, ist in Abbildung 2.17 die voraussichtliche Wachstumskurve der Things bis 2020 ersichtlich. MindSphere nennt diese Things auch Assets.

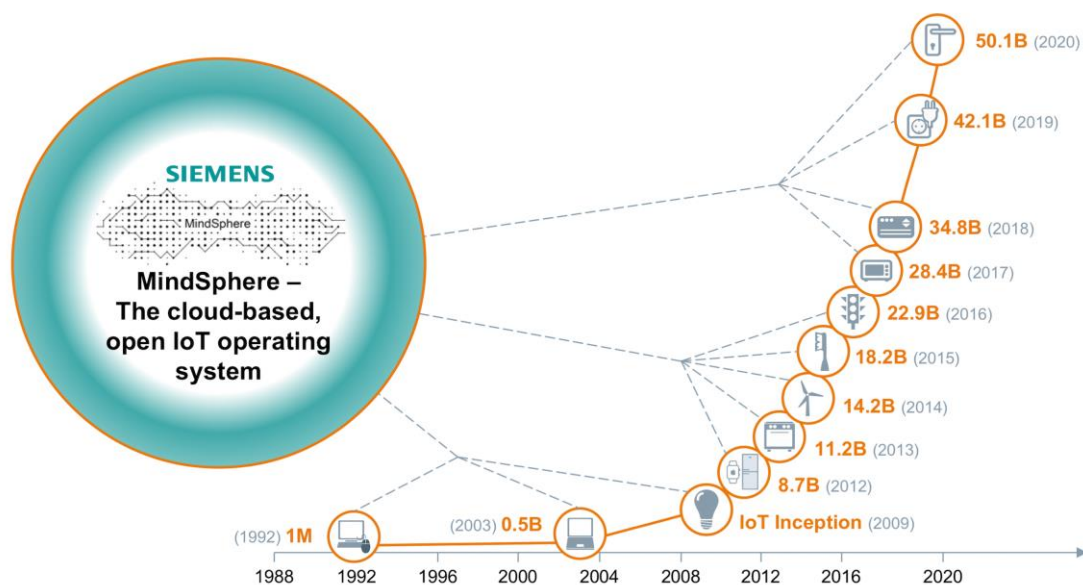


Abbildung 2.17: Wachstumsverlauf der Assets bis 2020 [23]

Diese Zahlen beruhen auf einer Studie der Digital Factory 2017. Dabei fand Siemens heraus, dass sich täglich rund 5,5 Millionen Things neu vernetzen. Für 2020 ist auf Basis dieser Studie eine Gesamtzahl von 50,1 Milliarden Things zu erwarten.

Die in der Grafik angegebene Einheit B steht für Billions.

2.4.1 Industrial Internet of Things

Spricht man von einer IoT Anwendung in einem industriellen Umfeld, nennt man dieses System auch Industrial Internet of Things (IIoT). Ausgangspunkt für das IIoT ist die sogenannte M2M (Maschine-to-Maschine) Kommunikation [24]. Darunter versteht man den automatischen Informationsaustausch von Endgeräten mittels Kabellösungen, Mobilfunkstandards oder anderen Übertragungstechnologien wie NFC, WLAN oder RFID [10].

Wird ein M2M-Netzwerk nun um die vielfältigen Funktionalitäten der Cloud erweitert, spricht man ebenso von einem IIoT (siehe Abbildung 2.18). Dabei werden beide Definitionen für das IIoT gebraucht. Vorteil dieses Systems, gegenüber dem M2M-Netzwerk, ist die viel universellere Möglichkeit, auf Daten zuzugreifen.

In der Cloud können Daten neben der simplen Speicherung auch analysiert, ausgewertet, für den weiteren Gebrauch aufbereitet und im gegebenen Fall sogar modifiziert werden [24]. Auch im Sinne einer Predictive Maintenance ergeben sich mit diesen Datenanalysen viele Möglichkeiten für die Optimierung.

In Zukunft sollte mittels Datenanalyse, ein Verbesserungsvorschlag oder ein Reparaturauftrag nicht nur vorgeschlagen, sondern im Sinne von Prescriptive Maintenance ausgeführt werden [25].

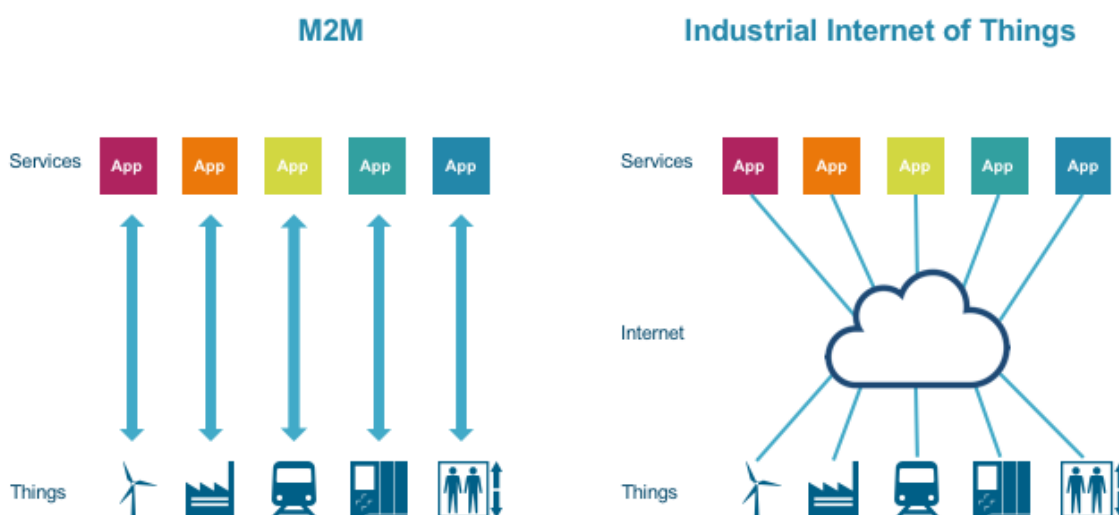


Abbildung 2.18: Vergleich von M2M und IIoT [26]

2.5 MindSphere

MindSphere ist ein von Siemens entwickeltes offenes Betriebssystem für Cloud Plattformen, das die Möglichkeit bietet, Daten aus Anlagen, Maschinen und Produktionssystemen zu analysieren und zu verarbeiten. Dabei handelt es sich um ein PaaS Modell, welches die Daten der Things aus dem IIoT nutzt. MindSphere bietet dabei eine Umgebung für Entwickler und App-Hersteller, die im PaaS-System ihre Anwendungen entwickeln und programmieren können [23].

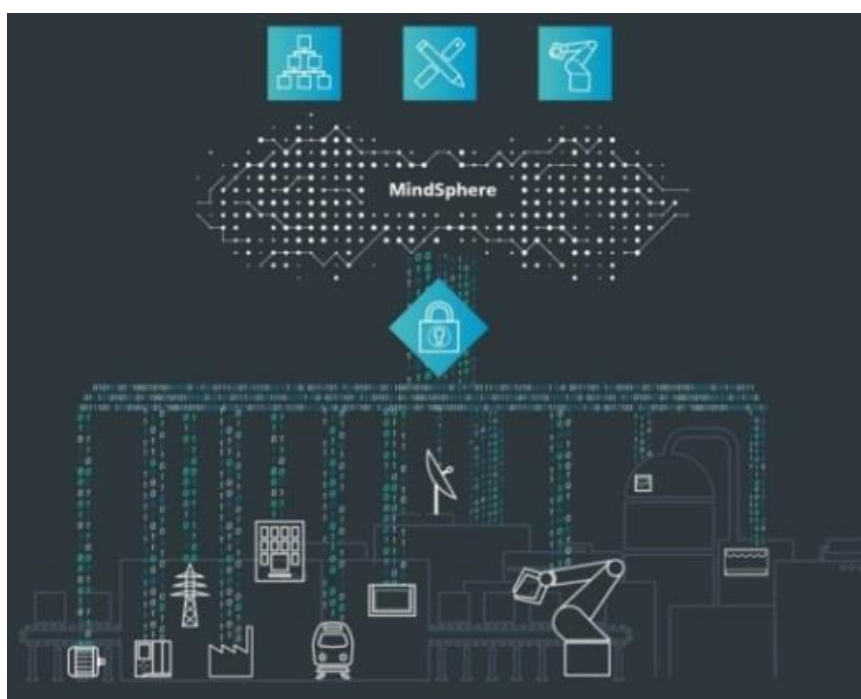


Abbildung 2.19: Symbolische Darstellung von Siemens MindSphere [23]

Die Motivation zur Entwicklung von MindSphere war die Verarbeitung von Produktions- und IIoT Daten, um die Effizienz zu steigern und Kosten zu senken.

Dabei sollten Daten aus allen Produktionsschritten genutzt und verarbeitet werden. Hierbei können diese Daten in einem Digital Twin oder generell für das Ganze PLM (Product Lifecycle Management) genutzt werden. Siemens will mit MindSphere ein Tool schaffen, das die Digitalisierung erleichtert. Zusätzlich ist Siemens Marktführer in der Automatisierungstechnik, was ein riesiges Potential für die Vernetzung und Erweiterung digitaler Dienste bietet [23].

2.5.1 Aufbau von MindSphere

Der Aufbau von MindSphere lässt sich anhand eines Schichtmodells erklären, welches aus drei Schichten aufgebaut ist (siehe Abbildung 2.20).

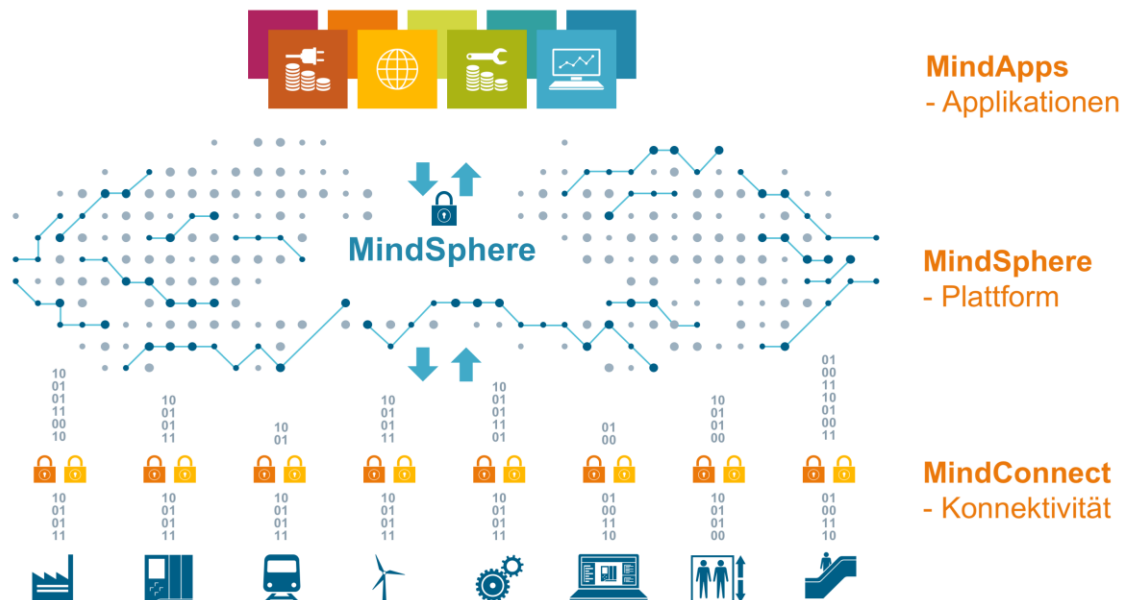


Abbildung 2.20: Drei-Schichten-Modell von MindSphere [23]

1. Konnektivitäts-Schicht

Diese befindet sich direkt an der Feldebene und verbindet diese mit Hilfe eines Gateways, dem sogenannten MindConnect, eine Steuerung oder einen Sensor mit der MindSphere Cloud. Kommunizieren kann die MindConnect mit den Protokollen S7 von Siemens, sowie mit dem OPC-UA (Open Platform Communications Unified Architecture) Protokoll. Mit Hilfe von OPC-UA lassen sich somit auch Fremdgeräte mit der Cloud verbinden.

Angeschlossen werden können dabei von Steuerungen bis zur Robotik alle Geräte die Daten im OPC-UA oder S7 Protokoll liefern [23] [26] [27].

Die MindConnect gibt es in zwei Hardwarevarianten, als MindConnect Nano und als MindConnect IoT 2040. Beide Geräte übertragen die oben angeführten Protokolle, unterscheiden sich aber in der Anzahl der Übertragungspunkte. Für das preiswertere MindConnect IoT 2040 Gateway, welches in Abbildung 2.21 ersichtlich ist, stehen max. 50 Datenpunkte zur Verfügung, wohin die MindConnect Nano, ersichtlich in Abbildung 2.22, max. 250 Datenpunkte anbinden kann.



Abbildung 2.21: MindConnect IoT 2040 [28]



Abbildung 2.22: MindConnect Nano [28]

MindConnect bietet folgende Vorteile [23]:

- Schnelle und einfache Anbindung an MindSphere
- Industrieprotokolle S7 sowie OPC-UA
- Software Updates im laufenden Betrieb
- Wartungsfrei und robustes Gehäuse
- Sicherheitskonzept nach Industriestandards
- Integrierter Datenpuffer, der Daten bei Verbindungsfehler mit der Cloud lokal am Gateway speichert.

Neben MindConnect Gateways gibt es weitere Möglichkeiten, eine S7-1500 SPS, mit der Cloud zu verbinden. Dazu muss ein eigener Funktionsbaustein im SPS Programm hinzugefügt werden. Dieses Thema wird in der Masterarbeit nicht

behandelt, wird hier aber zur Vollständigkeit aufgeführt. Vorteil des Funktionsbausteins ist der geringe Einsatz von Hardware. Genau dieser Kostenvorteil bringt jedoch den Nachteil, dass nur mehr Daten aus Siemens SPS gelesen werden können.

Eine zusätzliche Möglichkeit ist die Verbindung mit MindConnect LIB, eine auf der Programmiersprache C basierenden Bibliothek. Diese ermöglicht es Drittanbietern, mit Spezialanforderungen eine Verbindung mit der Cloud herzustellen [23] [26] [27].

2. Plattform-Schicht

Ist die Verbindung zu MindSphere hergestellt, kommt man auf die Plattformebene, also auf die Ebene, auf der sich die Cloud-Infrastruktur befindet. Diese wird je nach MindSphere Version von verschiedenen Herstellern bereitgestellt. In der in dieser Arbeit verwendeten MindSphere Version 2.0 wird die Cloud Infrastruktur von SAP genutzt. Die neue Version 3.0 wird voraussichtlich ab Sommer 2018 via Amazon Web Services zu Verfügung stehen [23] [26] [27].

3. App-Schicht

Die dritte Ebene, auch Applikationsebene genannt, ist die Ebene, auf der sich die sogenannten MindApps befinden, welche in HTML5 (Hypertext Markup Language) entwickelt werden. MindApps in diesem Kontext bieten Funktionalitäten wie, das Darstellen von Produktionsdaten. Aufrufbar sind sie über einen Browser wie Google Chrome oder den Internet Explorer [23] [26] [27].

2.5.2 Datentransport

Am Ende des theoretischen Teils wird der Datentransport vorerst nur prinzipiell erklärt, wie das Signal des Sensors zur App gelangt. Der genaue Weg der Daten wird in den folgenden Unterkapiteln erklärt. Zu Übersicht ist der Datentransport in Abbildung 2.23 ersichtlich und in den unten angeführten Punkten erklärt:

- Ein Sensor nimmt ein Signal auf, welches in der Folge von einem SPS Modul verarbeitet wird. Bei besagten SPS Modulen kann es sich um fehlersichere oder konventionelle Module handeln. Wird ein sicherheitsrelevantes Signal verarbeitet, kommt immer ein FSM zum Einsatz. Kommt ein Signal, wie zum Beispiel die CPU Temperatur, direkt aus dem FSM einer Baugruppe, wird dafür ein Datensatz benötigt. Dieser Datensatz wird von der SPS erstellt, um Daten aus den Modulen

auslesen zu können. Dabei handelt es sich nicht um Daten, die vom SPS Programm verarbeitet werden, sondern um Daten die direkt aus der Firmware der Module gelesen werden. Im Kapitel 3.1 Aufbau der SPS, wird genauer auf die Funktion und dem Aufbau eines Datensatzes eingegangen.

- Im nächsten Schritt gelangt das Sensorsignal von der dezentralen Peripherie in die F1500 CPU. Auf dieser läuft das Programm der Steuerung und zusätzlich werden hier die Datenübergangspunkte definiert.
- Von diesen Datenübergangspunkten aus wird das Signal von der SPS-CPU in das MindConnect IoT 2040 Gateway übertragen. Dieses ist mittels LTE Router verbunden, welcher die Verbindung zum Internet herstellt.
- MindSphere greift über den Router die Daten ab und verarbeitet diese in den vorinstallierten oder eigenständig entwickelten Apps.
- MindSphere ist nicht in der Lage, Steuer- und Regelungsaufgaben zu übernehmen, weil es dafür nicht entwickelt wurde. Daten können nur von der Feldebene in die Cloud geschickt werden. MindSphere selber kann die Daten analysieren und darstellen, aber nicht Befehle geben oder Aktionen ausführen lassen.

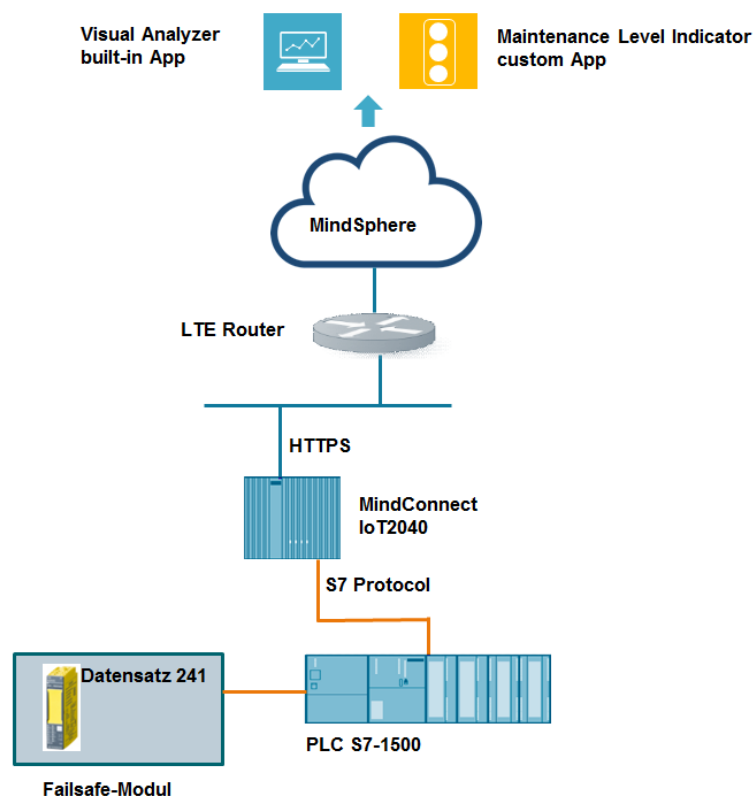


Abbildung 2.23 : Datentransport

3 Praktische Umsetzung des Demonstrators

Der praktische Teil der Masterarbeit wurde sowohl bei Siemens, als auch auf der Technischen Universität Graz umgesetzt. Die Programmierung der Steuerung als auch der Aufbau der SPS Hardware wurde größtenteils bei Siemens vorgenommen. Der Aufbau der restlichen Hardware erfolgte danach in der Smart Factory an der TU Graz. In diesem Kapitel wird der praktische Anteil der Arbeit beschrieben, das Programmieren der SPS Software, des Roboters, sowie auch der Aufbau und die Installation der ausgewählten Hardware.

3.1 Aufbau der Roboterzelle und Sensorik zur Datengenerierung

3.1.1 Unterbau

Als Unterbau für das mechatronische System wird eine Kombination aus Item und Festo Bauteilen verwendet. Der Rahmen des Aufbaus besteht aus Item Profilen, welche die Tischplatten von Festo tragen.

Das Grundgestell übernimmt dabei die Aufgabe, den Roboter und die Sensorik sicher und stabil zu tragen. Zusätzlich soll im Rahmen des Unterbaus die Steuerung des Roboters, sowie die Steuerung der Sensorik, als auch die Anbindung an MindSphere verstaut werden.

Außerdem ist es wichtig, dass der Unterbau verschiebbar ist bzw. mit Rollen ausgestattet wird, um eine Änderung des Einsatzortes zu ermöglichen.

Um die Mobilität zusätzlich zu erhöhen, wird der Roboter sowie alle Steuerungsteile über einen 230 V Netz betrieben. Ziel ist es, eine möglichst einfache Anbindung ans Stromnetz zu erreichen.

An der hinteren Seite des Unterbaus befinden sich zusätzlich Steher, welche die Aufgabe haben, Bildschirme zu tragen. Die Bildschirme dienen zur Visualisierung der MindSphere Daten. Ersichtlich ist die Konstruktion in der Abbildung 3.1.

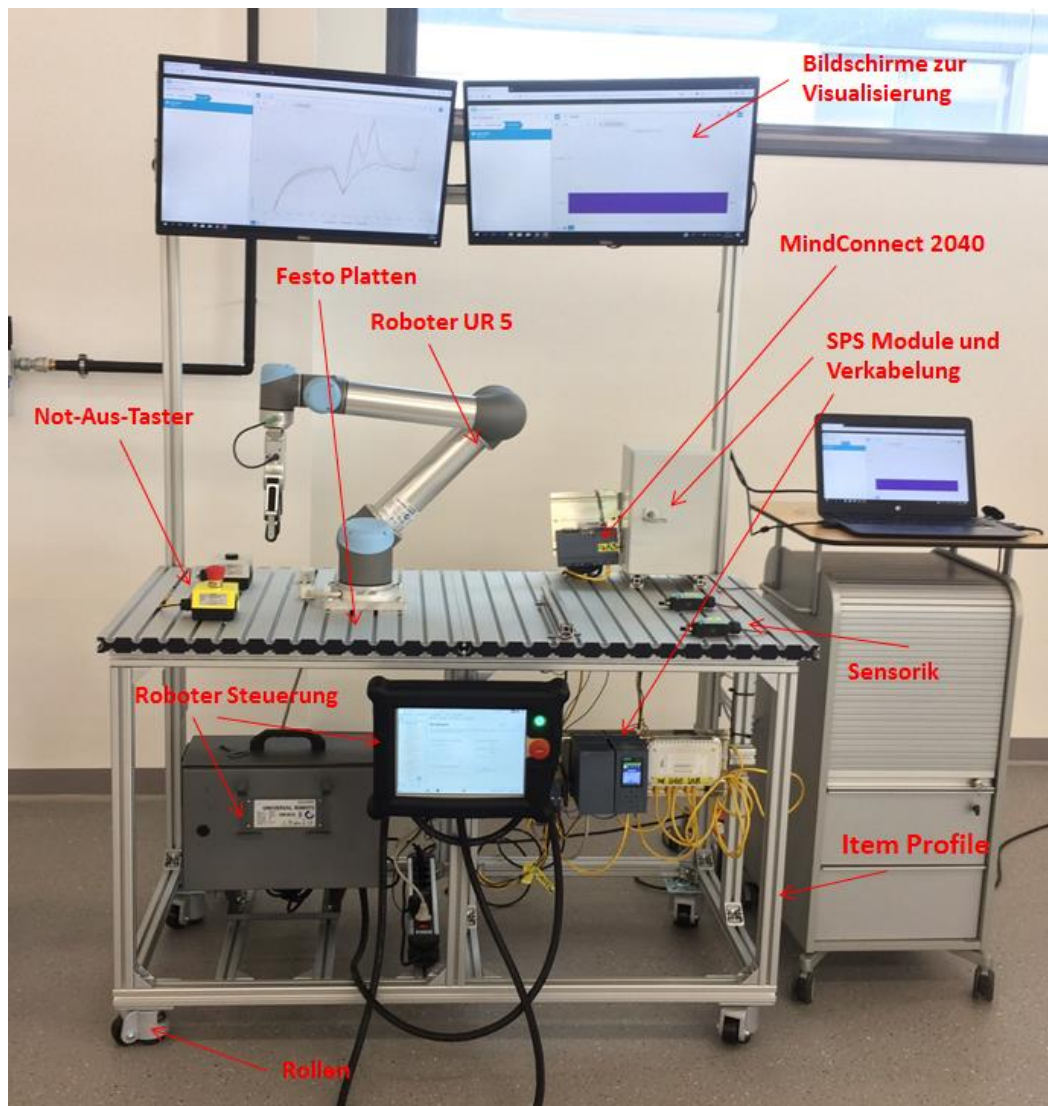


Abbildung 3.1: Gesamtaufbau

3.1.2 Roboter

Als Roboter für den Versuchsaufbau wird ein UNIVERSAL ROBOT vom Typ UR5 verwendet. Wie in Abbildung 3.2 ersichtlich, handelt es sich bei diesem Roboter um einen Sechssachsroboter.

Der UR5 ist vom Einsatztyp her ein collaborative robot, in der Kurzform hat sich die dazu die Bezeichnung „CoBot“ eingebürgert [29].



Abbildung 3.2: UR5 [30]

CoBots dürfen in unmittelbarer Nähe und mit Menschen zusammenarbeiten, ohne dass zusätzliche Schutzeinrichtung benötigt werden. Der UR5 benötigt daher keine klassischen Schutzeinrichtung wie z.B. Zäune, die den Menschen vor einem Zusammenstoß schützen [31].

Der Cobot verhindert Verletzungen, indem er die Motorströme der Antriebsmotoren überwacht. Kommt es im Falle eines Zusammenstoßes zu einem Überschreiten der Motorkraft und so zu einer erhöhten Stromaufnahme, wird der Roboter abgebremst und die Last reduziert [31]. Der betriebliche Einsatz von CoBots ist dennoch einer Reihe von sicherheitstechnischen Regularien unterworfen, auf die in dieser Arbeit jedoch nicht näher eingegangen wird.

Als Manipulator wird ein Greifer der Marke OnRobot (Abbildung 3.3) verwendet. Dieser lässt sich mit dem UR5 verbinden und arbeitet mit der Steuerung des Roboters zusammen.



Abbildung 3.3 On Robot Greifer

3.1.2.1 Montage

Der UR5 ist auf den Festo Profilplatten verschraubt. Dabei wurde die Grundplatte des Roboters mit vier M8 Schrauben an den zuvor angebrachten Nutensteinen in der Festo-Platte befestigt.

3.1.2.2 Programmierung

Der UR5 wird mit einem an der Steuerung befindlichen Tablett programmiert. Für jeden Demo-Case wurde dazu ein eigenes Programm angelegt, welches später zu einem großen zusammengefügt wurde, um den Roboter auch komplett selbständig laufen zu lassen. Die Programmierung an sich wird online, also direkt am Roboter durchgeführt. Das Gesamtprogramm sowie auch die einzelnen Unterprogramme werden zur Sicherung redundant auf der Steuerung gespeichert.

Programme für folgende Abläufe wurden erstellt:

- Schaltschranktüre öffnen:
Öffnet die Schaltschranktür für den Temperatur-Case.
- Not-Aus-Taster:
Verriegelt und Entriegelt den Not-Aus-Taster.
- Diskrepanz:
Zeigt den Demo-Case zur Darstellung von Diskrepanz-Unterschieden.
- Schaltschranktüre schließen:
Schließt die Schaltschranktür für den Temperatur-Case.
- Komplettlauf:
Gesamtprogramm, das die oben erwähnten Programme zusammenfasst.

3.1.3 Demo-Cases und Sensorik

Zu Beginn der Masterarbeit wurden Demo-Cases definiert, deren Hauptziel es war, die Kernkompetenz der Abteilung in den verschiedenen Demo-Cases darzustellen. Die Demo-Cases wurden daher auf Grund der Funktionen der FSM gewählt. Dafür wurde die Firmware der FSM so modifiziert, dass wie oben erwähnt, die Temperatur der CPUs und die Diskrepanzzeit von Signalen berechnet werden kann. Der Demo-Case für den Not-Aus-Taster ist noch nicht in die Firmware der FSM integriert und wird daher von einem SPS Programmbaustein verarbeitet. Sollte der Demo-Case „Not-Aus-Taster“ funktionieren, kann diese Integration ohne Probleme nachgeholt werden und einfach mittels Firmware Update auf die Module überspielt werden.

3.1.3.1 Not-Aus-Taster

Um diesen Demo-Case besser zu verstehen, wird zu Beginn ein Anwendungsbeispiel gegeben.

Anwendungsbeispiel:

Not-Aus-Taster müssen regelmäßig gewartet werden. Dabei spielen zwei Aspekte eine wichtige Rolle, einerseits die beschränkte Lebensdauer der Not-Aus-Taster und andererseits ihre vorgegebene regelmäßige Benutzung. Verantwortliche für die Sicherheitskomponenten in einem Werk haben das erste Mal über eine derartige Anwendung einen Überblick über die restliche Lebensdauer der Not-Aus-Taster, sowie über deren Benutzungsintervalle. Mit Hilfe dieser Daten kann eine effektive Wartung durchgeführt, gezielte Handlungen in Auftrag gegeben und die Gefahr von Verletzungen durch defekte Not-Aus-Taster reduziert werden.

Für diesen Versuchsaufbau wurde ein Not-Aus-Taster und ein Quittier-Taster über ein FSM an die SPS angeschlossen.

Der oben erwähnte UR5 betätigt mithilfe eines Manipulators die Taster und simuliert so eine Benutzung der Schalter, ersichtlich ist das in Abbildung 3.4.



Abbildung 3.4 Not-Aus-Taster

Diese generierten Daten werden über die MindConnect IoT 2040 an die MindSphere weitergegeben und dort mithilfe der MindApps verarbeitet.

Wie andere Bauteile unterliegen auch Not-Aus-Taster einer Lebensdauer, die in Zukunft überwacht werden kann. Für diesen Demo-Case wurde aus diesem Grund ein Zähler integriert, der die Lebensdauer mit Hilfe einer App in MindSphere überwachen kann. Kommt es zu einer Überschreitung der Lebensdauer, kann über MindSphere eine Warnmeldung kommuniziert werden.

Das zweite Problem, das ein Not-Aus-Taster mit sich bringt, ist die vorgeschriebene regelmäßige Benutzung der Taster. Mit Hilfe der Überwachung und Datenauswertung der Taster in MindSphere, kann das Nutzungsverhalten ersichtlich gemacht werden. Durch diese zusätzlichen Funktionen ergeben sich einige Vorteile für den Kunden. Neben der Verringerung der Gefahr durch den Ausfall von Not-Aus-Tastern, kann die Datenanalyse im Sinne von Predictiv Maintenance genutzt werden und so die Wartung effizient gestalten werden. Außerdem lässt sich mit den Stopp-Zeiten ein Produktionszeitverlust ermitteln.

3.1.3.2 Diskrepanzauswertung

Anwendungsbeispiel:

1. Ein Roboter legt Bleche in eine Bearbeitungsanlage ein. Dabei kontrollieren Taster, ob die Blechstreifen richtig positioniert wurden. Bei unterschiedlichen Auslösezeiten der Taster (=Diskrepanz), kann auf ein falsches Aufgreifen des Bleches, auf den Verschleiß der Greifer-Finger oder andere Ursachen geschlossen werden. Zusätzlich kann auch die Wiederholgenauigkeit des Roboters überprüft werden. Durch die Analyse der Daten kann im Sinne von Predictive Maintenance ein rechtzeitiges Tauschen von Komponenten erfolgen.

Ergänzend sei hier noch eine Idee für analoge Baugruppen (Diese befinden sich zum jetzigen Stand noch in der Entwicklung und dürfen daher nicht in den Versuchsaufbau integriert werden) angeführt:

2. Eine Zweisäulenpresse wird mit Kraftmessdosen ausgestattet, die die jeweiligen Kräfte an den Säulen messen. Mithilfe der FSM kann auch hier ein Unterschied in der Belastung ermittelt und so auf einen Verschleiß in einer der Säulen rückgeschlossen werden.

In der Masterarbeit wird die unterschiedliche Auslösezeit von Endlagenschaltern und Positionsschaltern überwacht. Für den Versuch erledigt ein Roboter (UR5) eine Pick & Place Aufgabe. Dabei wird ein Träger mithilfe eines Greifers auf die Positionsschalter gedrückt (Abbildung 3.5). Wird der Träger schief aufgegriffen, liefert dies zu weit auseinander liegende Auslösezeitpunkte der Positionsschalter.

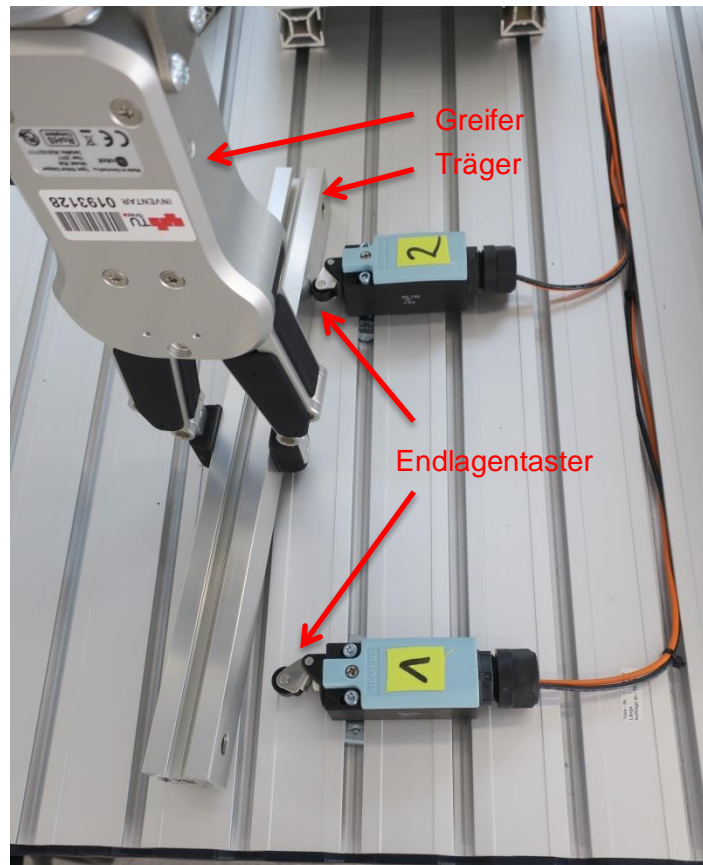


Abbildung 3.5 Demo-Case Diskrepanz

Diese Zeitdifferenzen werden über die FSM ermittelt und die Ergebnisse in MindSphere visualisiert. Die Zeiten selber werden in Prozent der vorgegebenen Zeit in MindSphere dargestellt. Ebenso kann eine Über- oder Unterschreitung dargestellt werden. Um sich den Verlauf der Daten besser vorstellen zu können, ist in Abbildung 3.6 der Datenfluss sichtbar. Zunächst werden im FSM die Diskrepanzwerte ermittelt und mit Hilfe eines Datensatzes gespeichert. Daraufhin werden die Daten der SPS-CPU (S7 1500) zur Verarbeitung übergeben, welche diese an die MindSphere App weiterleitet. Die Datensätze erklären MindSphere, wo im Programmcode der FSM die Diskrepanzwerte gespeichert sind. Das gleiche Prinzip wird auch beim Auslesen der CPU Temperaturen angewendet. Die genaue Funktion des Datensatzes wird in Kapitel 3.3 MindSphere Einrichtung beschrieben. Vorteil der Datenverarbeitung in MindSphere ist, dass die Daten über einen langen Zeitraum gespeichert und ausgewertet werden können. Mit Hilfe dieser Funktionen kann MindSphere einen Verlauf darstellen und somit ist es möglich, einen Trend zu erkennen. Bis dato besteht zwar die Möglichkeit die Daten in der SPS zu speichern, nur kann auf diese schwer zugegriffen werden und im Falle eines Stromausfalles kommt es zu einem Datenverlust.

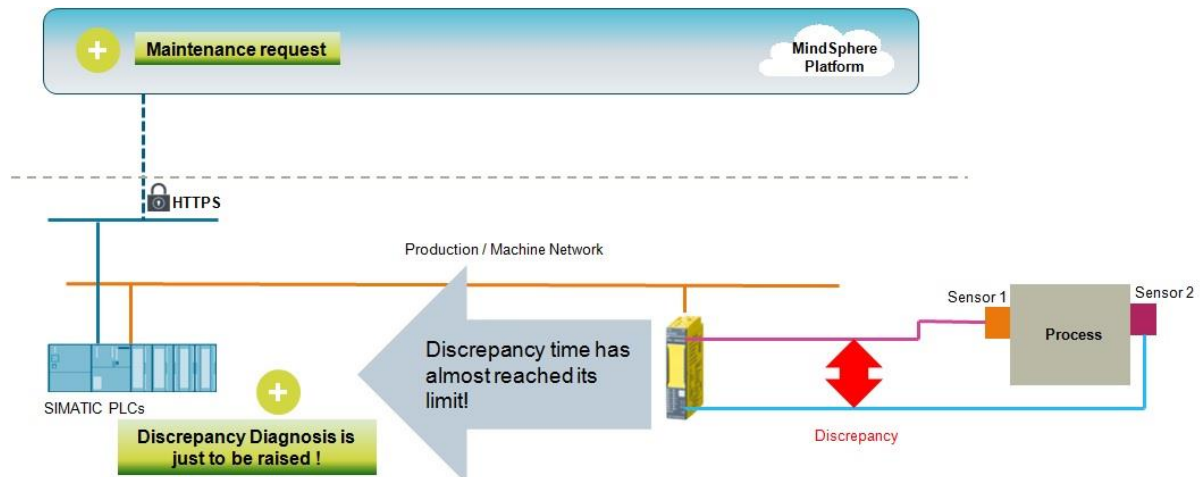


Abbildung 3.6: Datenfluss der Diskrepanz Sensorik

3.1.3.3 Temperaturüberwachung

Anwendungsbeispiel:

Die Module von speicherprogrammierbaren Steuerungen befinden sich in Schaltschränken. Viele dieser Schaltschränke haben aktive Heizungen, um Kondenswasser oder Schwitzwasser in den Schränken zu verhindern. Mithilfe einer Temperaturüberwachung der FSM kann ohne zusätzliche Sensorik auf Störungen in der Temperaturregelung geschlossen werden. Außerdem können andere Probleme, wie der Verschluss von Lüftungsschlitzen oder der Ausfall von Lüftern, über eine Temperaturänderung detektiert werden.

Für diesen Versuchsaufbau werden, wie in Abbildung 3.7 ersichtlich, mehrere FSM in einem Schaltkasten angeordnet. Zusätzlich werden die FSM mittels Heizfolie beheizt, um eine schnelle Erwärmung zu erreichen. Dabei übernimmt der Roboter das Schließen des Schaltkastens. Nach einer gewissen Heizzeit wird die Schaltschranktüre geöffnet und die Baugruppen werden gekühlt.

Die Heizung wird mittels eines Reed-Kontaktes gestartet, der in der Schranktür verbaut ist. Schließt dieser, geht ein Signal an die SPS, welche die Stromversorgung für die Heizung freigibt. Nach einer vorgegebenen Zeit oder, wenn die Tür wieder geöffnet, wird schaltet die Heizung automatisch ab.



Abbildung 3.7 Demo-Case Temperaturüberwachung

Wie in den Beispielen oben werden diese Temperaturwerte wieder in die MindSphere geschickt und dort über die Healthmonitor-App dargestellt. Beschrieben wird die App in Kapitel 3.3.4.2.2 Healthmonitor.

Ersichtlich ist der Aufbau in Abbildung 3.7.

Zusammenfassend bringt diese Analyse folgende Vorteile:

- a. Einfache Temperaturüberwachung aller FSM ohne zusätzliche Sensorik.
- b. Auslesen der Temperaturen und Darstellung des Temperaturverlaufes über die Zeit.
- c. Auskunft über die maximale Temperatur, um im Falle einer Überschreitung das Modul tauschen zu können.
- d. Rückschluss auf die Umgebungstemperatur bei gleichbleibenden Umwelteinflüssen.

3.2 Aufbau der speicherprogrammierbaren Steuerung

In diesem Kapitel wird der Aufbau der Steuerungskomponenten behandelt. Das Thema der MindSphere Anbindung inklusive der dafür benötigten Hardware wird in Punkt 3.3 behandelt.

3.2.1 Hardware

Siemens Standort Graz befasst sich mit der Entwicklung der FSM mit der Typebezeichnung ET200. Diese Module gehören der sogenannten „dezentralen Peripherie“ an. Bei diesen Baugruppen kann es sich um analoge und digitale Eingangsbaugruppen sowie auch um digitale und analoge Ausgabebaugruppen handeln. Speziell die Module für die ET200 wurden für den peripheren Einsatz entwickelt, darunter versteht man den Einsatz der Module nicht direkt an der CPU, sondern an einem weiter entfernten Standort. Diese Module haben den Vorteil, dass sie direkt bei der Sensorik oder der Maschine angebracht werden können. Nachteilig müssen die Module jedoch über eine Datenleitung wie z.B. PROFINET mit der CPU verbunden werden.

Um den Unterschied zu verdeutlichen, sieht man Abbildung 3.8 zuerst eine zentrale Peripherieeinheit, die direkt auf der S71500 CPU hängt, also mit dem Rückwandbus verbunden ist. Darunter in Abbildung 3.9 ist die dezentrale Peripherieeinheit zu erkennen, die mittels eines gelben PROFINET-Kabels an die CPU angeschlossen wird.

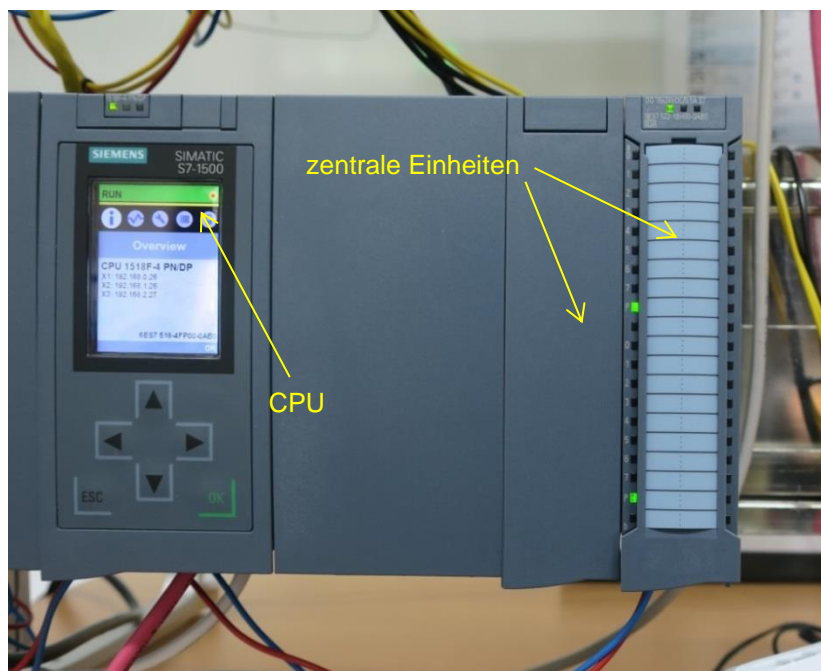


Abbildung 3.8 : Zentrale Peripherie

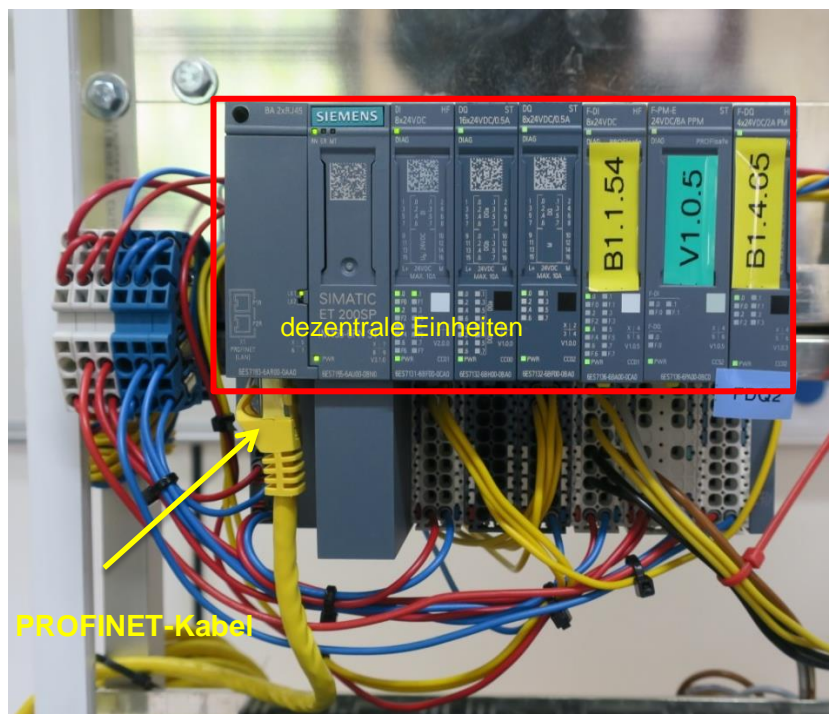


Abbildung 3.9 : Dezentrale Peripherie

In Tabelle 1 sind die verwendeten Bauteile der SPS. In der Tabelle stehen neben der Positionsnummer (P.NR), die Bauteilbezeichnung, die Artikelnummer sowie die Stückzahlen.

P.NR	Bauteilbezeichnung	Artikelnummer	Stück
1	Laststromversorgung PM 190W, 120/230V AC, 24 V DC, 8 A	6EP1333-4BA00	1
2	F-DI 8x24VDC HF	6ES7136-6BA00-0CA0	5
3	F-DQ 8X24VDC/0.5A PP	6ES7136-6DC00-0CA0	2
4	M 155-6 PN HF inkl. Servermodul	6ES7155-6AU00-0CN0	2
5	Bus Adapter 2xRJ45	6ES7193-6AR00-0AA0	2
6	BU-Typ A0, 16 Push-In, 2 Einspeiseklemmen Gebrückt (Digital-/Analog, 24VDC/10A)	6ES7193-6BP00-0BA0	7
7	BU-Typ A0, 16 Push-In, 2 Einspeiseklemmen Getrennt (Digital-/Analog, max.24VDC/10A)	6ES7193-6BP00-0DA0	2
8	Servermodul	6ES7193-6PA00-0AA0	2
9	DQ 8x24VDC/2A HF	6ES7132-6BD20-0CA0	1
10	AQ 4xU/I ST	6ES7135-6HD00-0BA1	1
11	CPU 1516F-3 PN/DP	6ES7516-3FN01-0AB0	1

12	Memory Card, 256 Mbyte	6ES7954-8LL03-0AA0	1
13	LAN und Stromkabel für die gesamte SPS-Einheit		
14	Profilschienen für die Montage	6ES7590-1AB60-0AA0	1

Tabelle 1: Tabelle der verwendeten Komponenten

Um im Sinne der Dokumentation die Verständlichkeit zu erhöhen, wird in der folgenden Auflistung kurz die Funktion der jeweiligen Bauteile beschrieben, darunter befinden sich noch Bilder, um die Bauteile leichter erkennen zu können.

1. Versorgt die gesamte Steuerung mit Strom.
2. F-DI steht für fehlersichere digitale Inputbaugruppe, diese Baugruppe kann digitale Signale aufnehmen und weiterverarbeiten, z.B. die Aufnahme eines Signals eines Endlagensensors.
3. F-DQ steht für fehlersichere digitale Ausgangsbaugruppe, diese Baugruppe kann digitale Single versenden, z.B. zur Ansteuerung eines Relais.
4. Interfacemodule dienen dazu, die dezentralen Module mit der CPU zu verbinden. Sie verbinden den Busadapter und die Einspeiseklemmen.
5. Der Busadapter wird auf die Servermodule gesteckt und ermöglicht eine Anbindung der dezentralen Peripherie an die CPU per PROFINET.
6. Einspeiseklemmen dienen dazu, die Eingangs- und Ausgangssignale an die Module weiterzuleiten. Dabei gibt es 2 Varianten, die gebrückten und die getrennten Einspeiseklemmen. Die gebrückten Einspeiseklemmen können die Stromversorgung der nebenliegenden Einspeiseklemmen verwenden, wogegen bei der getrennten Variante jede Klemme einzeln mittels Kabel mit Strom versorgt werden muss.
7. Siehe Punkt 6.
8. Servermodule dienen zum Schließen des Rückwandbusses eines SPS Aufbaus, sie dienen als Endstücke und sitzen daher immer rechts am Ende des letzten Moduls.
9. Wie Punkt 2, nur nicht fehlersicher.
10. Wie Punkt 3, nur nicht fehlersicher.
11. Die CPU ist der zentrale Bestandteil, auf dieser läuft das Programm, welches zuvor im TIA Portal entstanden ist.
12. Die verwendete CPU 1500 muss mit einer Speicherkarte betrieben werden, auf ihr speichert die CPU das Programm.

In Abbildung 3.10 ist der Aufbau jener Modulgruppe ersichtlich, die aus den in Tabelle 1 gelisteten Komponenten besteht.

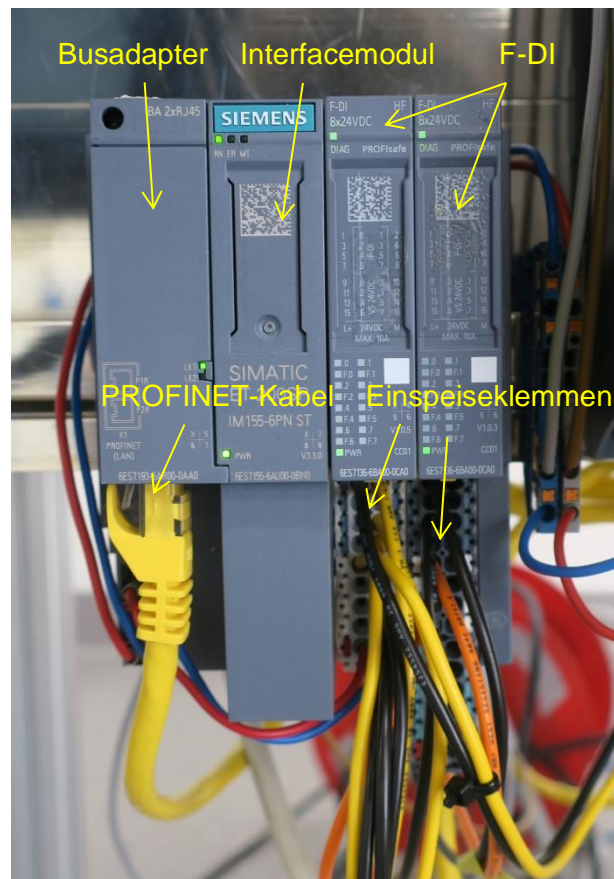


Abbildung 3.10: Aufbau einer Moduleinheit

In diesem Aufbau werden mittels eines Switchs zwei dezentrale Peripherieeinheiten an die CPU angeschlossen. Zusätzlich hängt auf diesem Switch noch die Verbindung zum MindConnect IoT 2040 Gateway, welches wiederum mit einem LTE Router verbunden ist. Die erste dezentrale Peripherieeinheit befindet sich im Unterbau und trägt später im Programm den Namen „MindSphere_2“ (Abbildung 3.11), die andere dezentrale Peripherieeinheit befindet sich wie oben erwähnt im Schaltschrank für den Temperatur Demo-Case und trägt später in der Software den Namen „MindSphere_1“ (Abbildung 3.7).

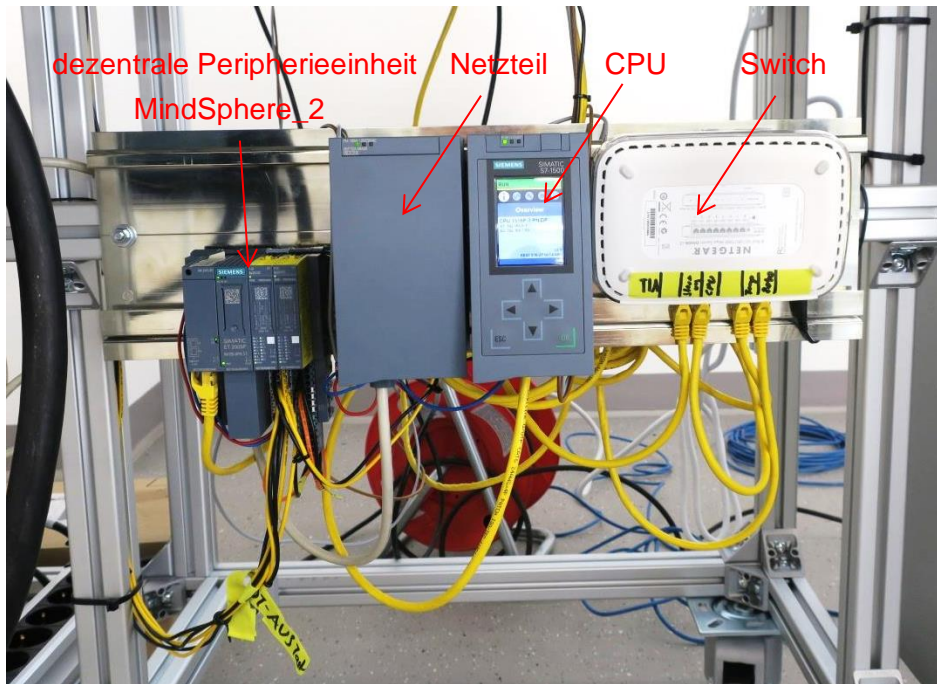


Abbildung 3.11: SPS-Hardware Unterbau

Wie zuvor beschrieben, führt ein PROFINET-Kabel zum MindConnect Gateway, und eines vom Gateway zum LTE Router bzw. zu einem anderen Netzwerk mit Internetverbindung. Ersichtlich ist dies in Abbildung 3.12.

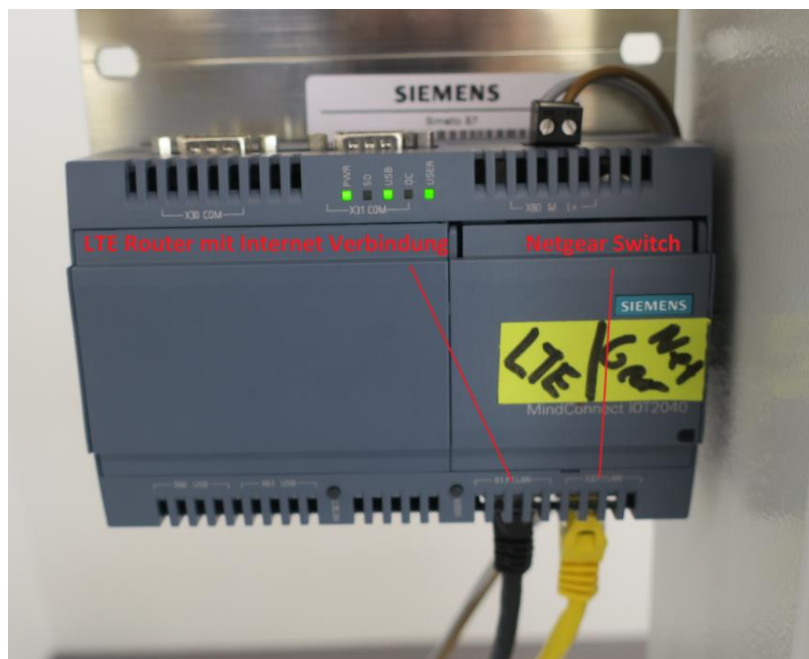


Abbildung 3.12: MindConnect IoT 2040

3.2.2 Software

Die Software für die Steuerung wurde in der Siemens Programmiersprache Step 7 (auch S7) geschrieben. Zur Programmierung verwendet man dazu das Totally Integrated Automation Portal (TIA-Portal) von Siemens.

Das TIA Portal dient dazu, alle Automatisierungs- und Planungsaufgaben durchzuführen. Für diese SPS Programmierung wird die TIA Version V15 inklusive dem Safety Paket zur Programmierung sicherheitsrelevanter Schaltungen verwendet. STEP 7 ist dabei ein Teil des TIA Portals, welches für die Programmierung der SPS verantwortlich ist.

Im TIA Portal stehen wiederum mehrere verschiedene Programmiersprachen zur Verfügung. In diesem TIA Projekt befindet sich die Hochsprache SCL (Structured Control Language) sowie die grafische Programmiersprache FUP (Funktionsplan) wie sie in Abbildung 3.13 ersichtlich sind. Links sind zwei UND und eine ODER Verknüpfung zu sehen, welche grafisch als FUP dargestellt werden. Rechts sieht man einige Zeilen eines klassischen SCL Codes.

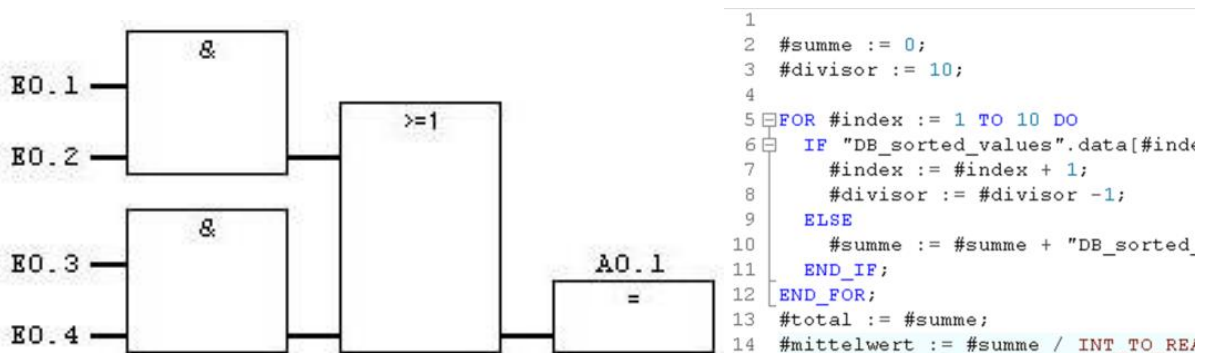


Abbildung 3.13: Gegenüberstellung von FUP und SCL [32] [33]

In den nächsten Schritten wird erklärt, wie das SPS Programm aufgebaut ist, Hardware hinzugefügt wird, welche Softwareteile für die Verarbeitung der Daten verantwortlich sind und wie die Daten von der MindSphere App abgefragt werden können. Weiters wird erläutert, wie die Datenpunkte verknüpft und für die richtige Informationsübergabe eingestellt werden.

3.2.2.1 Hardwarekonfiguration

Im ersten Schritt wurde die verwendete Hardware im TIA Portal festgelegt. Wie in Abbildung 3.14 ersichtlich wird eine CPU und 2 dezentrale Peripherieeinheiten verwendet. Die im Bild linke Peripherieeinheit übernimmt die Steuerung der Sensorik, die rechte Peripherieeinheit wird für den Temperatur Demo-Case verwendet. Genauer sind die einzelnen Teile in Abbildung 3.15 und Abbildung 3.16 ersichtlich.

In Abbildung 3.14 sieht man die verwendete CPU, die mit einem PROFINET Kabel (grün) mit den Einheiten MindSphere 1 und 2 verbunden sind.

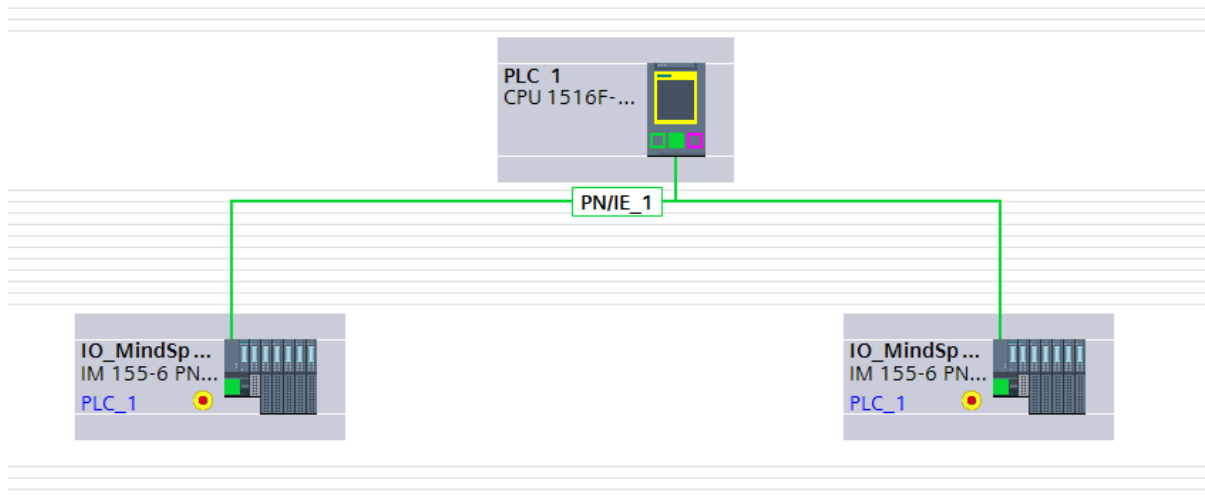


Abbildung 3.14: Netzwerkplan und Grundaufbau der SPS

In Abbildung 3.15 ist die detaillierte Ansicht der dezentralen Peripherie von MindSphere_2 ersichtlich. Zu sehen sind die verwendeten Einheiten für die Überwachung der Diskrepanz und des Not-Aus-Tasters.

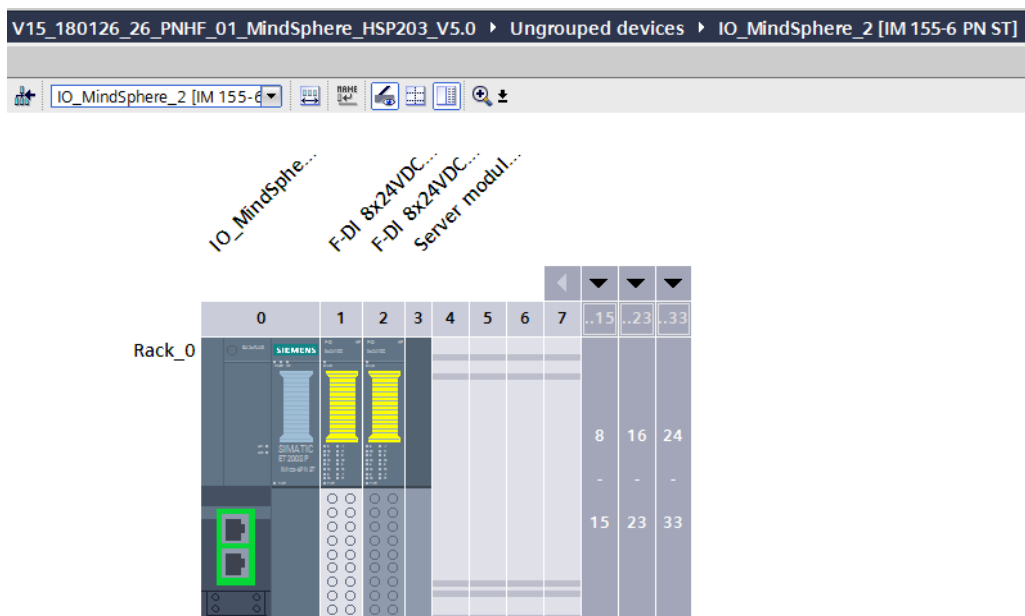


Abbildung 3.15: Dezentrale Peripherie MindSphere 2

In Abbildung 3.16 ist die dezentrale Pheripherie MindSphere_1 für den Temperatur Demo Case ersichtlich. Wie bereits erwähnt, übernehmen diese Module grundsätzlich keine Steuerung von Prozessen, die hohe Anzahl an Modulen wurde gewählt, um die Temperaturentwicklung zu begünstigen, zusätzlich wird die Stromversorgung der Heizung vom Modul 7 übernommen. Einem DQ Modul mit 24 V Spannung und einer maximalen Stromstärke von 2 A.

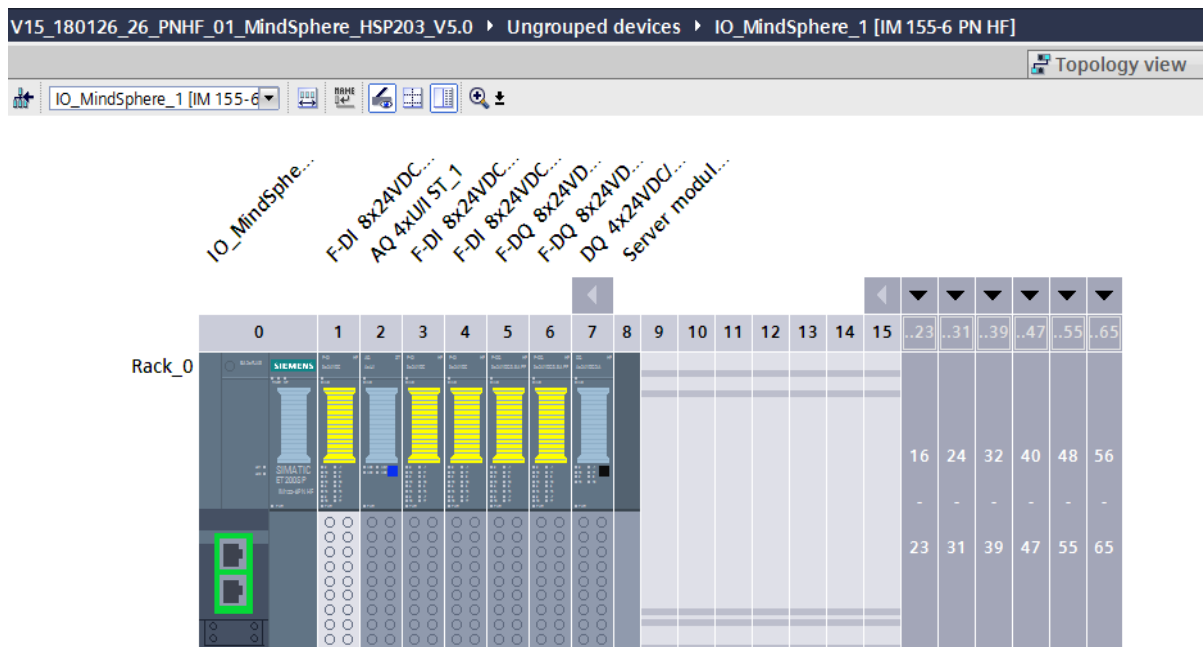
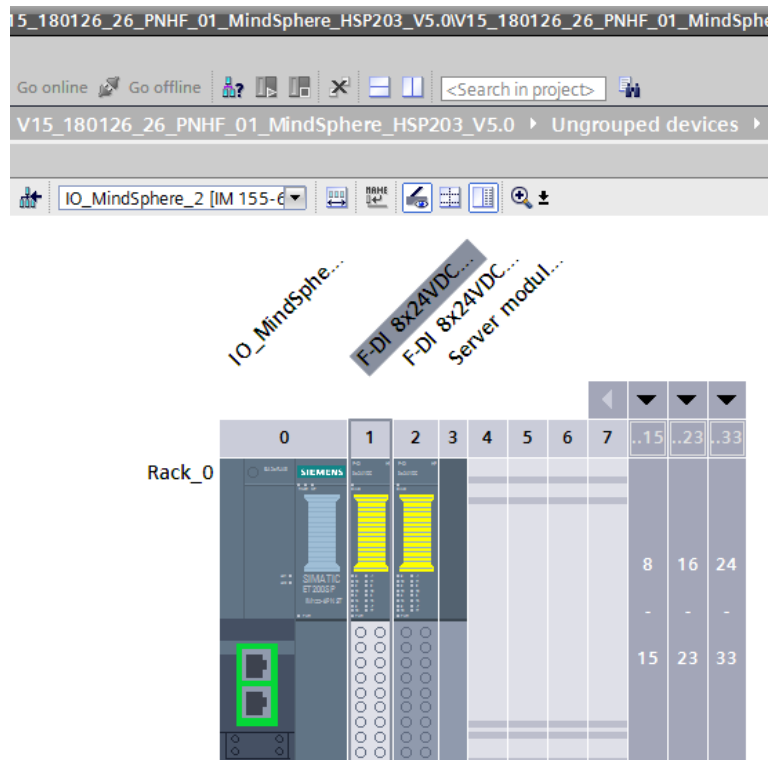


Abbildung 3.16: Dezentrale Peripherie MindSphere 1

Bei den hinzugefügten Module müssen nun die Ein- und Ausgänge benannt werden. Am einfachsten geht das, indem man mit einem Doppelklick auf das Modul klickt und unter IO tags für die verschieden Eingänge Namen zuweist. In Abbildung 3.17 ist als Beispiel eine F-DI 8x24 ersichtlich. F_DI_0_0 ist dabei der Name für den ersten Eingang. F steht dabei für Fail-Safe Modul, DI für digitalen Input, 0 für das erste Modul in der SPS und die zweite 0 für den ersten Eingang im Modul. Die Namen ergeben sich so numerisch aufsteigend. Für die verschiedenen Anwendungen muss jeder Ein- und Ausgang benannt werden, um diesen später im Programm nutzen und zuweisen zu können. Ist der Name wie z.B. F_DI_0_0 nun vergeben, lässt sich dieser später einfach beim Programmieren aufrufen und erscheint dadurch im Programmbaustein.



F-DI 8x24VDC HF_1 [F-DI 8x24VDC HF]					
General		IO tags	System constants	Texts	
Name	Type	Address	Tag table	Comment	
F_DI_0_0	Bool	%I26.0	Default tag table		
F_DI_0_1	Bool	%I26.1	Default tag table		
F_DI_0_2	Bool	%I26.2	Default tag table		
F_DI_0_3	Bool	%I26.3	Default tag table		
F_DI_0_4	Bool	%I26.4	Default tag table		
F_DI_0_5	Bool	%I26.5	Default tag table		
F_DI_0_6	Bool	%I26.6	Default tag table		
F_DI_0_7	Bool	%I26.7	Default tag table		

Abbildung 3.17: Zuweisung von Zugängen

Wird ein sicherheitsrelevantes Bauteil wie ein Not-Aus-Taster angeschlossen, muss nach dem in der Theorie beschriebenen Prinzip 1oo2 gearbeitet werden.

Genau dieses muss auch in der Hardware der Module eingestellt werden und verursacht so ein Zusammenlegen der Kanäle.

Vom Not-Aus-Taster kommen insgesamt 4 Kabel zum Modul, da dieser in sich redundant aufgebaut ist. Im Not-Aus-Taster sind zwei gleiche Schalter integriert, um im Falle eines Defekts die Not-Aus Funktion garantieren zu können. Zwei dieser Kabel gehen auf den Eingang 1 und 5, also wie oben beschrieben F_DI_0_1, F_DI_0_5, die zwei anderen gehen auf die dazugehörigen Massestellen der Einspeiseklemmen, in der das Modul steckt. Somit ist der Stromkreislauf geschlossen. Um eine 1oo2 Auswertung zu bekommen, müssen im

TIA-Portal wieder mittels Doppelklick auf das Modul die Einstellungen geöffnet werden (Abbildung 3.18). Unter „General“ findet man die „Channel Parameters“. Öffnet man diese, kann man die Einstellungen für die jeweiligen Kanäle ändern. In diesem Fall hängt der Taster auf dem 1 und 5 Eingang. Diesen wählen wir aus und ändern unter „Sensor Evaluation“, von 1001 auf 1002. Im Feld „Discrepancy“ kann noch die Zeit eingestellt werden, ab welcher die Baugruppe einen Fehler ausgibt. Werden also die beiden Eingänge verglichen und unterscheiden sich im Eingangssignal, wartet die Baugruppe in diesem Fall 5ms ab und meldet dann einen Fehler. Korrigiert sich der Prozess innerhalb der vorgegebenen Zeit, meldet die Baugruppe keinen Fehler.

Angenommen, eines der Kabel am Not-Aus-Taster bricht, so ist das 1002 Prinzip nicht mehr erfüllt und die Baugruppe geht in den fehlersicheren Zustand. Unter einem fehlersicheren Zustand ist der Zustand zu verstehen, der Auftritt, wenn an einem FSM ein Fehler anliegt. Im Unterschied zu einem standard Modul erkennt ein FSM Ausfälle und Fehler und löst darauf hin Reaktionen, die zuvor im SPS-Programm definiert wurden, aus.

Für die unterschiedlichen Channels, in diesem Fall der erste und der fünfte, lassen sich noch zusätzliche Einstellungen machen. Auswählen kann man dabei die Quelle des Signals, den „sensor supply“. Dabei unterscheidet man ob die Massekabel wie in unserem Fall am gleichen Modul liegen, das heißt, ob der Kreislauf geschlossen ist, oder ob ohne Masseanschluss nur ein Signal in die Baugruppe geschickt wird. Diese Einstellungen müssen wiederum für beide Kanäle getätigt werden.

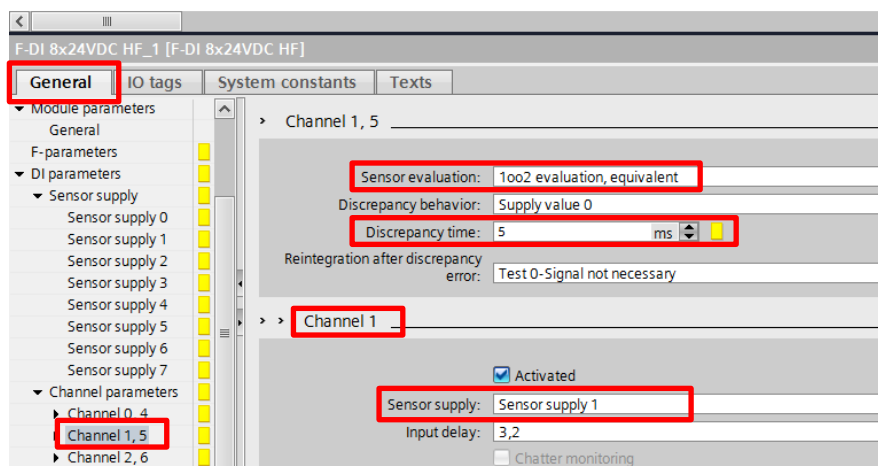
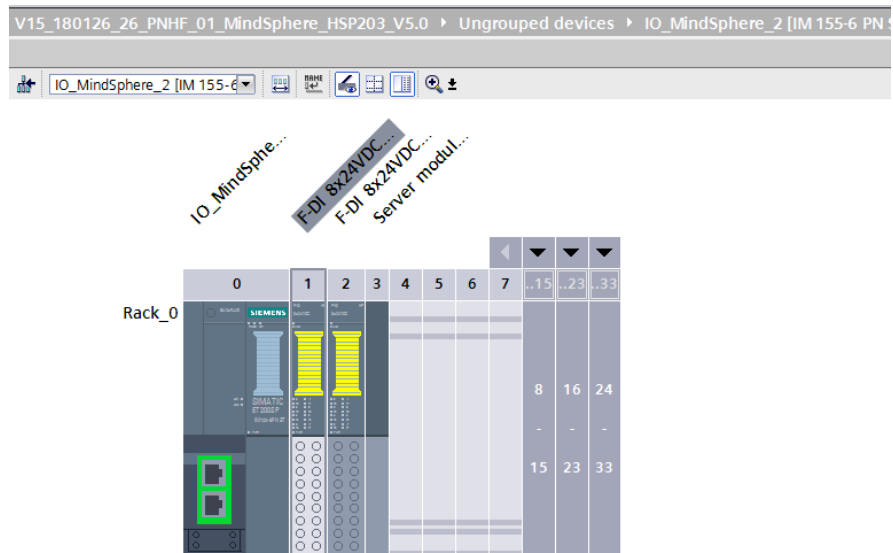


Abbildung 3.18: Channel Settings

Die gleichen Einstellungen müssen für die Sensoren der Endlagentaster gewählt werden. Diese müssen diese gleich wie die Not-Aus-Taster verkabelt werden. Dazu kommen auch hier wieder vier Kabel von zwei Schaltern zu den Modulen. Der Anschluss funktioniert wieder gleich wie beim Not-Aus-Taster. Die Einstellungen ändern sich hingegen. In diesem Fall ist eine lange Diskrepanzzeit möglich. Endlagensensoren müssen nicht in 5ms reagieren, sondern erst nach einer anderen vorgegeben Zeit, im Falle des Demonstrators z.B. 5 Sekunden. Zusätzlich könnte man diesen Fall erweitern, indem man analoge Signale verarbeitet. Dabei könnte man nicht die Zeit einstellen, sondern zum Beispiel einen Schwankungsbereich für Temperaturwerte. Würde ein Fall außerhalb der vorgegebenen Diskrepanz auftreten, zum Beispiel ein Sensor auf Grund eines Fehlers mehr oder weniger als erlaubt anzeigen, würde die Baugruppe einen Fehler melden. Auch hier ließen sich wieder Vorhersagen für die Zukunft treffen lassen, indem das Auseinanderdriften von Werten beobachtet wird.

Um die Dokumentation vollständig zu halten, sind in Abbildung 3.19 noch die Einstellungen für den Demo-Case mit der Endlagensensorik dargestellt. Grundsätzlich werden die gleichen Einstellungen verwendet, nur wurden diesmal die Taster an die Channels null und vier gelegt und die Diskrepanzzeit auf 5 sec. (= 5000ms) angehoben.

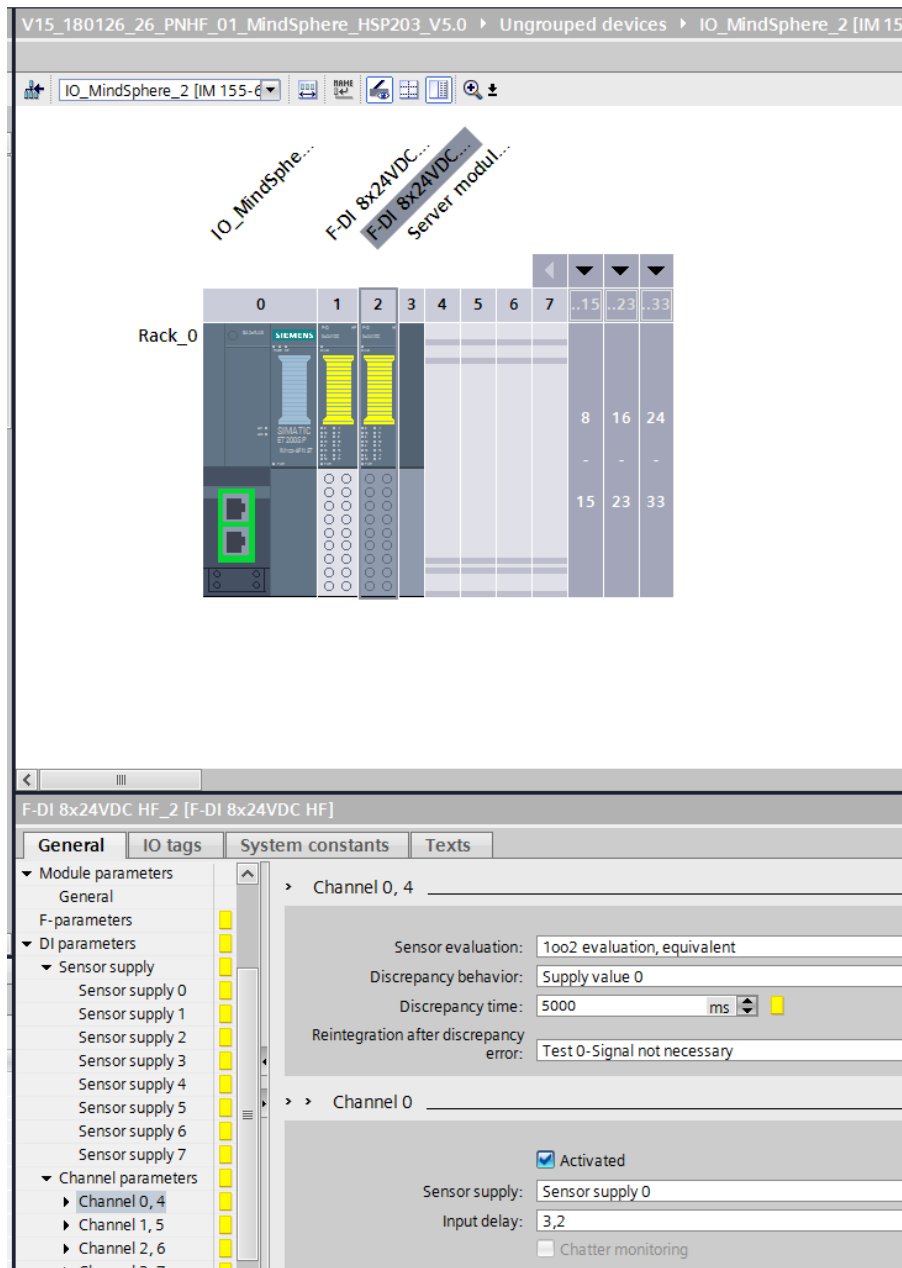


Abbildung 3.19: Einstellungen Diskrepanz-Case

3.2.2.2 SPS Bausteine

In diesem Punkt werden die benötigten Programmblöcke besprochen. Zuvor wird noch definiert, was ein Organisationsbaustein (OB), ein Funktionsbaustein (FB), eine Funktion (FC) und ein Datenbaustein (DB) ist.

Der Grundbaustein wird OB genannt, er ist die Verbindung zum Betriebssystem der CPU und wird, wie in Abbildung 3.20 ersichtlich, zyklisch abgearbeitet. Ohne OB kann das Programm nicht auf der CPU laufen. Um die Programmierung zu strukturieren, werden im OB Funktionen und Funktionsbausteine aufgerufen, in denen sich Teilprogramme der Steuerung befinden. Wird ein FB oder eine FC nicht im OB aufgerufen, wird diese auch nicht ausgeführt. FB besitzen im Unterschied zu FC einen zugeordneten Speicherbereich, einen sogenannten Datenbaustein. Spricht man von einem DB der einem FB zugehörig ist, bezeichnet man diesen Datenbaustein als „Instanz DB“. Auf diesen Instanz DB kann über den Aufruf des FB zugegriffen werden. Ein FB kann auch mehr als nur einen zugeordneten Instanz DB besitzen [33] [1].

Datenbausteine enthalten Daten und keine Programme wie OB, FC oder FB, welche auch als Codebausteine bezeichnet werden können. DB können Daten wie Stückzahl, Uhrzeit oder andere beliebige Daten enthalten, die die SPS verarbeiten kann. Daher ist es für die Bausteine sehr wichtig, den richtigen Datentyp festzulegen, um so eine Weiterverarbeitung zu ermöglichen.

Bei Funktionen handelt es sich um Bausteine, für die kein Speicherbereich vorgesehen ist, sprich die Daten gehen nach der Bearbeitung verloren und können nur über sogenannte Merker im System gespeichert werden. Zu erwähnen ist noch, dass sowohl FB in FC als auch FC in FB aufgerufen werden können. Wie oben beschreiben, werden die Bausteine zyklisch abgearbeitet. Unter zyklischer Abarbeitung versteht man die Abarbeitung der Bausteine im OB von oben nach unten und in weiterer Folge von links nach rechts, ersichtlich in Abbildung 3.20 [33] [1].

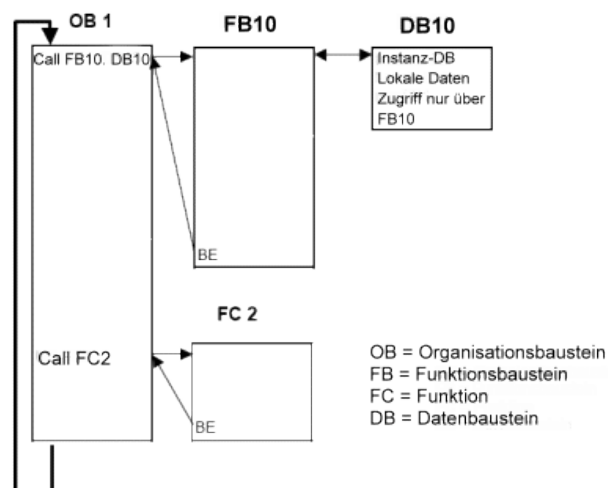


Abbildung 3.20: Zyklische Durchlauf der Bausteine [33]

Mit diesem theoretischen Wissen kann das Programm nun im TIA Portal leichter erklärt werden. Die violett gefärbten Bausteine im Programm sind OB, die grünen Blöcke sind FC, die hellblauen Blöcke sind die oben erwähnten FB und die dunkelblauen Blöcke deren Instanz DB. Da sich in unserer Steuerung Safety Komponenten befinden, gibt es diese Bausteine für das Safetyprogramm ebenfalls. Diese sind gelb hinterlegt und am unteren Ende der Abbildung 3.21 ersichtlich.

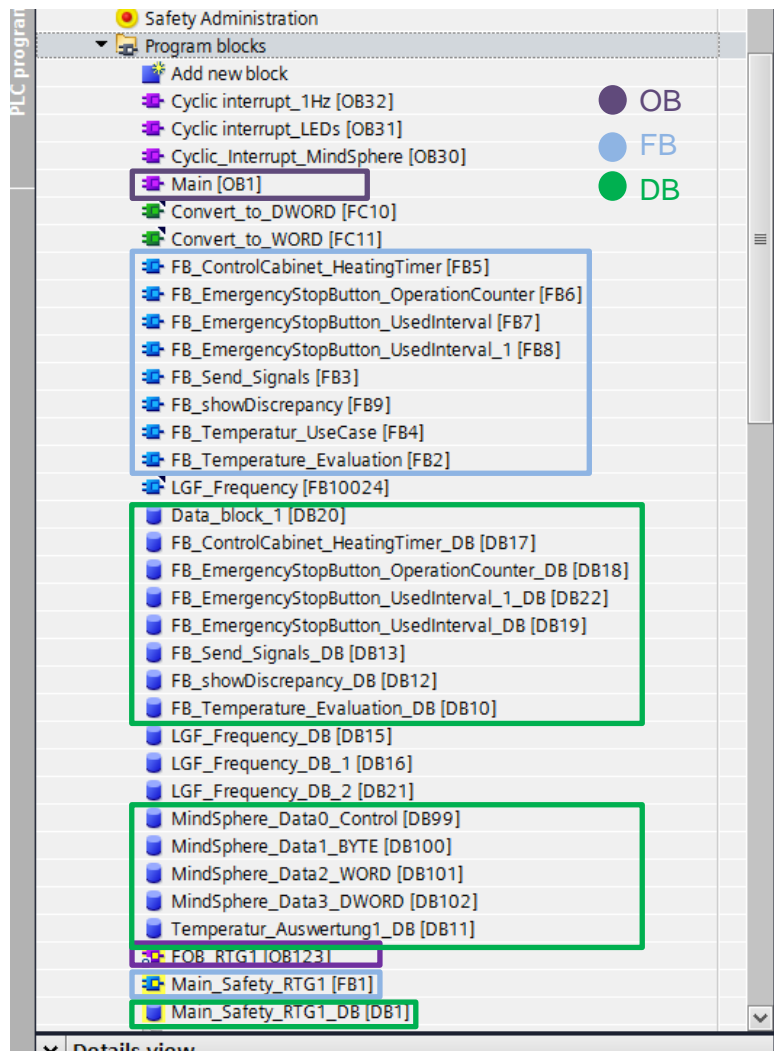


Abbildung 3.21: Im Programm verwendete Bausteine

Das für die Masterarbeit verwendete Programm wurde auch für eine andere MindSphere Anbindung verwendet, deshalb befinden sich Bausteine im Programm, die für dieses Projekt nicht genutzt werden, z.B. der OB32 und OB31. Diese Bausteine wurden in einem anderen Test verwendet, um Signalleuchten und die Leuchtdioden bei Fehlern an den Modulen blinken zu lassen. In den nächsten Schritten werden die einzelnen, für dieses Projekt verwendeten Bausteine genauer beschrieben.

3.2.2.2.1 OB

In diesem Punkt werden die für das Programm der Masterarbeit verwendeten OB aus Abbildung 3.21 erklärt.

1. OB1:
 - Im OB1 oder auch Main OB werden die von uns verwendeten FB aufgerufen, und zyklisch abgearbeitet. Die verwendete Sprache in diesem Baustein ist SCL. In
 -
 - Abbildung 3.22 sind die aufgerufenen FB ersichtlich. Der grün hinterlegte Code dient als zusätzliche Option zu dem darunter befindlichen FB. Genauer wird das in den nächsten Punkten erklärt.

IF...	CASE... OF...	FOR... TO DO...	WHILE... DO...	(*...*)	REGION
1					
2					"FB_Temperature_Evaluation_DB" ();
3					
4					"FB_Send_Signals_DB" ();
5					
6					"FB_ControlCabinet_HeatingTimer_DB" ();
7					
8					"FB_EmergencyStopButton_OperationCounter_DB" ();
9					
10					/"FB_EmergencyStopButton_UsedInterval_DB" (#out_time);
11					
12					"FB_EmergencyStopButton_UsedInterval_1_DB" (pushButton=>#pushButtonAlarm, ...);
15					
16					"FB_showDiscrepancy_DB" (inRangeValue=>#outDIs, ...);
18					

Abbildung 3.22: OB 1

2. FOB_RTG1

Dieser OB dient zum Aufruf der Safety Funktion Bausteine.

3.2.2.2.2 FB

In diesem Punkt werden die FB Bausteine aus Abbildung 3.21 erklärt.

1. FB_ControlCabinet_HeatingTimer (Abbildung 3.23):

Der erste Baustein übernimmt die Steuerung des Heizelements. TP steht für Impuls Timer, dieser Baustein generiert nach einem Eingangsimpuls ein Signal „Counted_Time“. Eingang ist dabei der Kanal „F_DI_0_0“, also der erste Eingang des ersten SPS Moduls. Ist dieser Eingang beschalten startet der „Counted_Time“ „Value“ abzulaufen. Der Ausgang „Counted_Time“ ist so lange true (1), solange die Zeit (130.000ms) läuft. Ist diese abgelaufen, schaltet der Baustein auf false (0). Mit dem Ausgang „Counted_Time“ wird die Funktion „HeatOn“ angesteuert, welche den

Ausgang DQ_1_3 auf true (1) schaltet. Ist nun „DQ_1_3“ true, zieht das Heizelement für den Temperatur Demo-Case Strom aus dem DQ Modul.

Der 3. Baustein dient dazu die Heizung abzuschalten, wenn der Eingang „F_DI_0_0“ auf false (0) wechselt. Dies geschieht durch die Negierung des Eingangs, ersichtlich durch einen Kreis neben der Beschriftung. Der letzte Baustein ist nur zum Generieren eines Impulses da, er lässt ein Blinklicht leuchten. Dieses kann z.B. dazu verwendet werden, um den Stromfluss im Heizelement anzuzeigen.

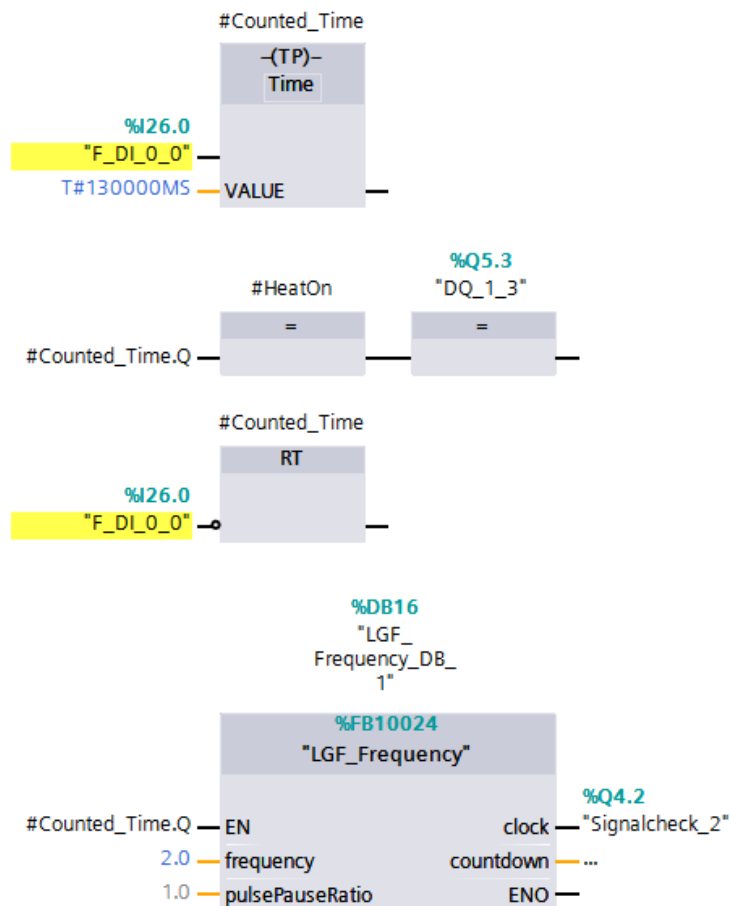


Abbildung 3.23: FB_ControlCabinet_HeatingTimer

2. FB_EmergencyStopButton_OperationCounter (Abbildung 3.24):

Dieser Baustein zählt die Betriebszyklen des Not-Aus-Tasters. Realisiert wurde das mit dem Baustein „CTU“, welcher in Abbildung 3.24 ersichtlich ist. „CU“ entspricht dem Counter, dieser erhöht den Wert „CV“ bei jedem neuen Signal am Eingang.

„PV“ steht für den Grenzwert, welcher den Counter auslöst. In unserem Fall 0, da bei Digitalwerten nur 0 und 1 auftreten können. Wäre ein analoges Signal vorhanden könnte der Grenzwert so eingestellt werden, dass nur Werte ab einer bestimmten Größe verarbeitet werden.

„Q“ steht für den Ausgang. Der zweite Baustein löst beim Überschreiten einer gewissen Anzahl von Betätigungen ein Signal aus, welches dem System zeigt, dass

der Schalter getauscht werden sollte. Im Fall der Abbildung 3.24, wurde diese Anzahl an Betätigungen auf den Wert 5 gesetzt. Kommt ein Signal zum Eingang „F_DI_0_7“ also zu „R“, wird der Counter auf null gesetzt. Dies dient als Reset Funktion für den Zähler.

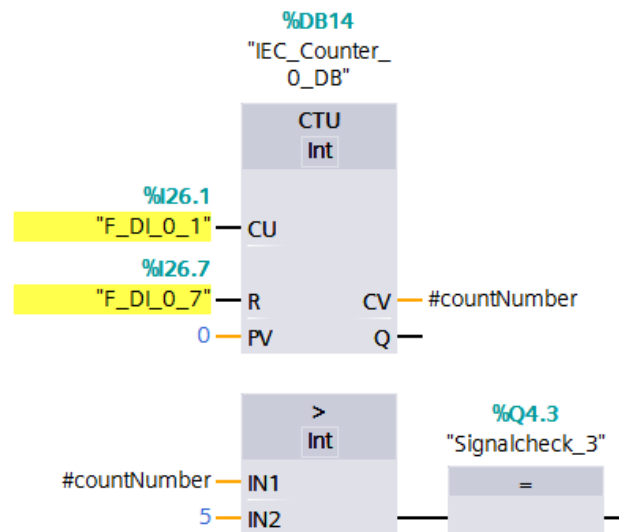


Abbildung 3.24: FB_EmergencyStopButton_OperationCounter

3. FB_EmergencyStopButton_UsedInterval (Abbildung 3.25):

Hier gibt es zwei 2 Bausteine. Der erste Baustein wurde in der Hochsprache SCL programmiert, welche weniger wartungsfreundlich ist. Daher beschreiben wir den Baustein hier in der FUP Sprache.

Der FB hat die Aufgabe, die Zeitdifferenz zwischen den letzten Benutzungen darzustellen, einen Zeitstempel zu generieren und im Falle einer Überschreitung ein Warnsignal auszusenden. RD_LOC_T ermittelt die aktuelle Zeit inklusive Datum und gibt diese über outputLocTime aus. Mit Hilfe des Bausteins T_DIFF können zwei Zeitstempel subtrahiert werden. Daraus ergibt sich der Wert TimeDiff in Millisekunden.

Saved_Time ist die Zeit, die im dritten Baustein das letzte Mal gespeichert wurde.

Dieser dritte Baustein bekommt zu Beginn ein Eingangssignal F_DI_0_1, welches negiert wurde, ersichtlich ist das durch den runden Kreis davor. Der Eingang ist immer True(1) wenn der Schalter nicht gedrückt ist. Mit der P_Trig Funktion wird das Signal so lange gespeichert bis ein Signal F_DI_0_1 kommt. Diese Funktion nennt man in der Fachsprache auch Flankenauswertung. Mit dem Befehl Move ändert das Programm die outputLocTime in die Saved_Time.

Das Programmverhalten lässt sich wie folgt zusammenfassen. Ist der Taster nicht gedrückt, speichert der dritte Baustein die Zeit, an der der Taster das letzte Mal gedrückt wurde in Saved_Time. Über die aktuelle Zeit wird danach die Differenz

gebildet, welche als TimeDiff ausgegeben wird. Im letzten Baustein wird die letzte Zeit mit dem vorgegebenen Benutzungsintervall verglichen, um im Falle eines Überschreitens ein Warnsignal verschicken zu können.

Diese beschriebene Funktion befindet sich momentan auf der SPS, in Zukunft sollen diese Funktionen nach Möglichkeit von einer App berechnet oder direkt in die Firmware der Module integriert werden.

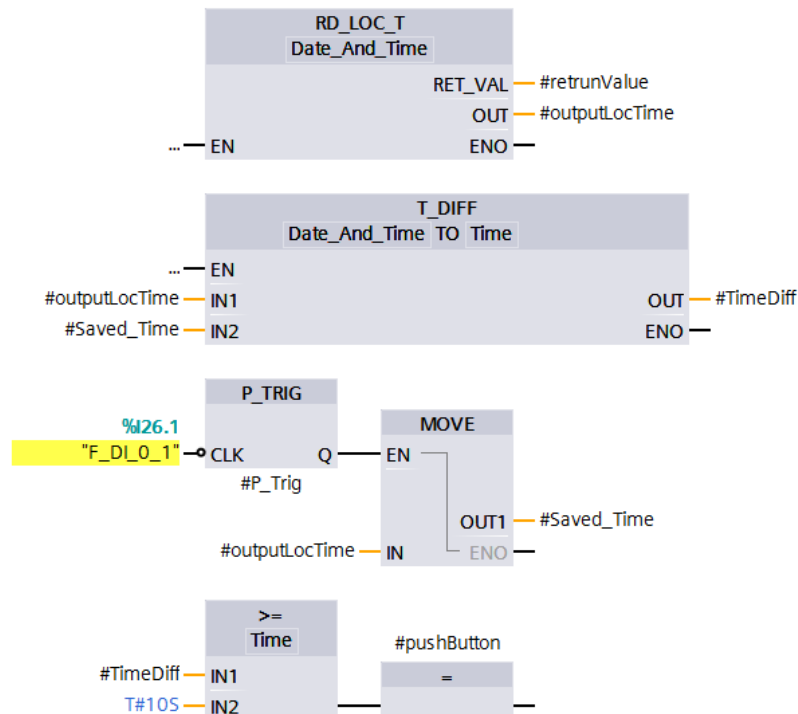


Abbildung 3.25: FB_EmergencyStopButton_UsedInterval

4. FB_Send_Signals

In diesem Baustein werden Signale von einer Baugruppe zur einer anderen geschickt, um eine Last zu erzeugen und so die Module im Schaltschrank zu erhitzen.

5. FB_showDiscrepancy (Abbildung 3.26):

Dieser Funktionsbaustein soll die Zuweisung der Diskrepanzwerte darstellen. Dabei spielen drei Werte eine wichtige Rolle. Ist die Diskrepanz innerhalb eines vorgegeben Wertes, spricht man von einem „InRangeValue“, ist er außerhalb bekommen wir den zweiten Wert, einen „outRangeValue“. Der dritte Wert stellt dar, in welchem der beiden Bereiche sich dieser Wert befindet.

Der Datenpunkt „MindSphere_Data2_WORD“.Sources[1].Elements[11]“ gibt genau diesen dritten Wert an. Ist dieser 1 wird der erste Baustein abgewickelt, ist dieser 2 wird der zweite Baustein durchgelaufen. Der weitere Ablauf der Bausteine ist

derselbe, nur läuft das Programm einmal für den „InRangeValue“ und einmal für den „outRangeValue“. Als Beispiel wird der erste Baustein beschrieben.

Entspricht der Eingangswert „MindSphere_Data2_WORD“.Sources[1].Elements[11]“ der Anzahl 1, ist die erste Bedingung erfüllt. Mit der darauf folgenden Move Funktion wird der Zahlenwert aus dem „MindSphere_Data2_WORD“.Sources[1].Elements[9]“ Datenpunkt ausgelesen und dem „InRangeValue“ zugewiesen. Im nächsten Schritt wird der „outRangeValue“ auf null gesetzt, um zu vermeiden, dass die Werte doppelt angezeigt werden. Das gleiche Schema funktioniert auch mit dem zweiten Baustein, somit erhält man immer nur einen Wert je nachdem ob, man sich im zulässigen Bereich der Diskrepanzzeit befindet oder nicht.

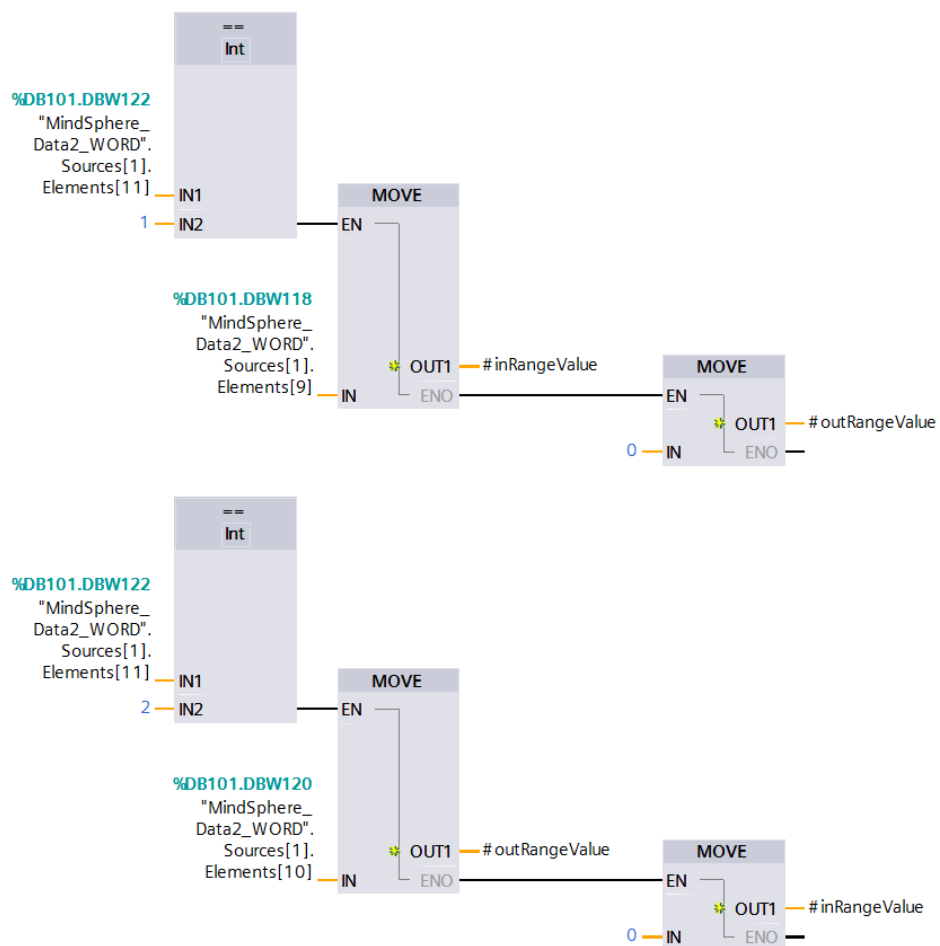


Abbildung 3.26: FB_showDiscrepancy

6. Die FB 4 und 2 sind nicht Teil dieses Projekts

7. Main_Safety_RTG1 (Abbildung 3.27):

In diesem Baustein muss die F-DI 8x24 aufgerufen werden. Diese Baugruppe benötigt diesen Aufruf, um nicht zu Beginn in einen Not-Stop zu gehen. Dieser Aufruf muss für jedes dieser Module einzeln programmiert werden. Exemplarisch wird der

Code für das erste Modul gezeigt. Erwähnt wird dieser Punkt, um einen Fehler beim Hinzufügen neuer Module zu vermeiden. Entscheidend ist, dass DQ Baugruppen nicht aufgerufen werden dürfen, da diese an sonst einen Fehler aufweisen.

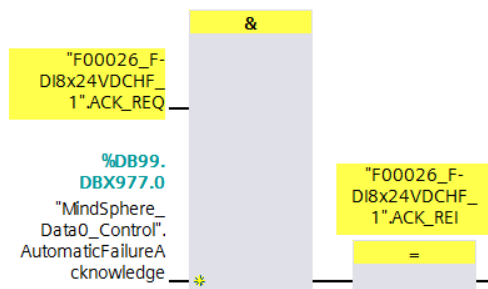


Abbildung 3.27: Main_Safety_RTG1

3.2.2.2.3 DB

Die in den Datenbausteinen gespeicherten Werte können von MindSphere abgerufen werden. Daher wird in diesem Kapitel gezeigt, wo die Übergabepunkte sind und wie diese festgelegt werden, um einen Datentransport zu ermöglichen.

Zuvor sollte jedoch noch ein Datenbaustein (Abbildung 3.28) ohne MindSphere Anbindung gezeigt werden. Im Datenbaustein befinden sich in der Spalte „Namen“ alle Variablen, die im Projekt verwendet wurden. Aufgeteilt sind diese zum Beispiel in „Input“, „Output“, „InOut“ Variablen. Diese beschreiben, ob es sich um Eingangs- oder Ausgangsvariablen handelt. Im Unterpunkt der jeweiligen Variablen wird dann noch der Name vergeben, welcher in anderen Unterprogrammen aufgerufen werden kann.

In der nächsten Spalte steht der Datentyp. Dieser hängt von dem im FB verwendeten Programmcode ab. Wird zum Beispiel ein Schalter projektiert und der Input mit dem Namen „HeatOn“ einem digitalen Eingang zugewiesen, erscheint im DB automatisch der Datentyp Bool. Zusätzlich kann noch ein Startwert angegeben werden.

FB_ControlCabinet_HeatingTimer_DB			
	Name	Data type	Start value
1	Input		
2	Output		
3	HeatOn	Bool	false
4	InOut		
5	Merker	Bool	false
6	InOut_1	Int	0
7	Static		

Abbildung 3.28: Datenbaustein ohne MindSphere Anbindung

Im nächsten Schritt sollen die Datenbausteine für die Datenübermittlung (Abbildung 3.29) erklärt werden. Dabei unterscheiden wir zwischen zwei Varianten. Erstens, Daten bei denen direkt aus einem SPS-Code eines FB ein Datenwert übermittelt werden soll und zweitens

Daten, die direkt von der Firmware der FSM kommen. Die zweite Variante ermöglicht es, die Daten abzugreifen, ohne dass dafür eine FB oder ein Programm geschrieben werden muss. Voraussetzung hierfür ist, dass die FSM diese Funktionen bereits implementiert haben. Siemens integriert die Anwendung einmal und der Kunde kann so ohne weiteren Programmieraufwand die Daten beziehen, ohne im TIA einen FB zu erstellen, der diese Daten auswertet und übergibt. Vorhanden ist die zweite Variante für die Diskrepanz und die Temperaturmessung, nicht jedoch für den Not-Aus-Taster.

Für die erste Variante muss der Datenbaustein jedoch richtig vorbereitet werden. Dazu muss im Datenbaustein unter „Properties“ und „Attributes“ der Haken bei „Optimized Block Access“ entfernt werden. Diese Option ordnet den Daten automatisch Speicherplatz im Baustein zu. Dies soll vermieden werden, da MindSphere wissen muss wo genau die Daten liegen, um auf diese zugreifen zu können. Entfernt man diesen Haken, erscheint wie in Abbildung 3.29 ersichtlich das Layout für den Datenbaustein mit der zusätzlichen Spalte „Offset“.

FB_EmergencyStopButton_UsedInterval_1_DB (snapshot created: 4/5/2018 3:53:41 PM)							
	Name	Data type	Offset	Start value	Retain	Accessible f...	V
1	Input				<input type="checkbox"/>	<input type="checkbox"/>	
2	Output				<input type="checkbox"/>	<input type="checkbox"/>	
3	pushButton	Bool	0.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	InOut				<input type="checkbox"/>	<input type="checkbox"/>	
5	TimeDiff	Time	2.0	T#0s	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	P_Trigger	Bool	6.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
7	outputLocTime	Date_And_Time	8.0		<input type="checkbox"/>	<input type="checkbox"/>	
8	Static				<input type="checkbox"/>	<input type="checkbox"/>	
9	retrunValue	Int	14.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Saved_Time	Date_And_Time	16.0	DT#1990-01-01	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Abbildung 3.29: Datenbaustein mit MindSphere Anbindung

Über den Offset kann genau ermittelt werden, wo die Daten abgelegt werden. Die Zahlen, die in der Offsetspalte stehen, werden automatisch erstellt. Je nach Datentyp wird die Speicherplatzgröße von selbst zugewiesen. „TimeDiff“ startet beim zweiten Bit und benötigt vier Bit an Speicherplatz, daher kann der nächste zu speichernde Wert erst bei 6.0 starten.

Erfolgt das Auslesen mithilfe einer auf einem Modul befindlichen Firmware, benötigt MindSphere auch hier den genauen Ort und einen Datensatz, der beschreibt wo die Daten abgelegt werden. Dazu gibt es ein Excel File, welches beschreibt, wo die Daten für die Temperatur oder die Diskrepanzwerte liegen. Diese beiden Excel Datenblätter befinden sich im Anhang. Zu unterscheiden ist hierbei wieder je nach Modul, also ob ein digitales oder ein analoges FSM Verwendung findet. Spricht man von einem F-AI Modul, also einem FSM mit analogem Input, benötigen man den Datensatz 241. Dieser wird in dieser Masterarbeit nicht verwendet, da Siemens die F-AI noch nicht freigibt. Für das F-DI Modul, also dem Fail-Safe Modul für digitale Eingänge, wird der Datensatz mit dem Namen 201 verwendet.

Dazu wird am einfachsten an Hand eines Beispiels erklärt, wie man den Datensatz ausliest und wie der Weg der Daten definiert wird. Als erstes wird die Hardware Identifikationsnummer benötigt. Diese lässt sich einfach aus dem TIA-Portal auslesen, dazu öffnet man sein Projekt, klickt auf „Device configuration“ und danach mittels Doppelklick auf das Modul, welches verwendet werden soll. In unserem Fall nehmen wir das F-DI Modul der dezentralen MindSphere_2 Einheit. Ersichtlich ist diese aus dem oben erwähnten Netzwerkplan. Hier soll die Temperatur als auch die Diskrepanz zweier Sensoren ausgelesen werden. Nach dem Doppelklick sollte wie in Abbildung 3.30 der Reiter „System constants“ erscheinen, Klicken man auf diesen erhält man unter „Hardware identification“ die Hardware ID welche automatisch vom TIA Portal zugewiesen wird. In diesem Fall die Nummer 301, welche im nächsten Schritt benötigen wird.

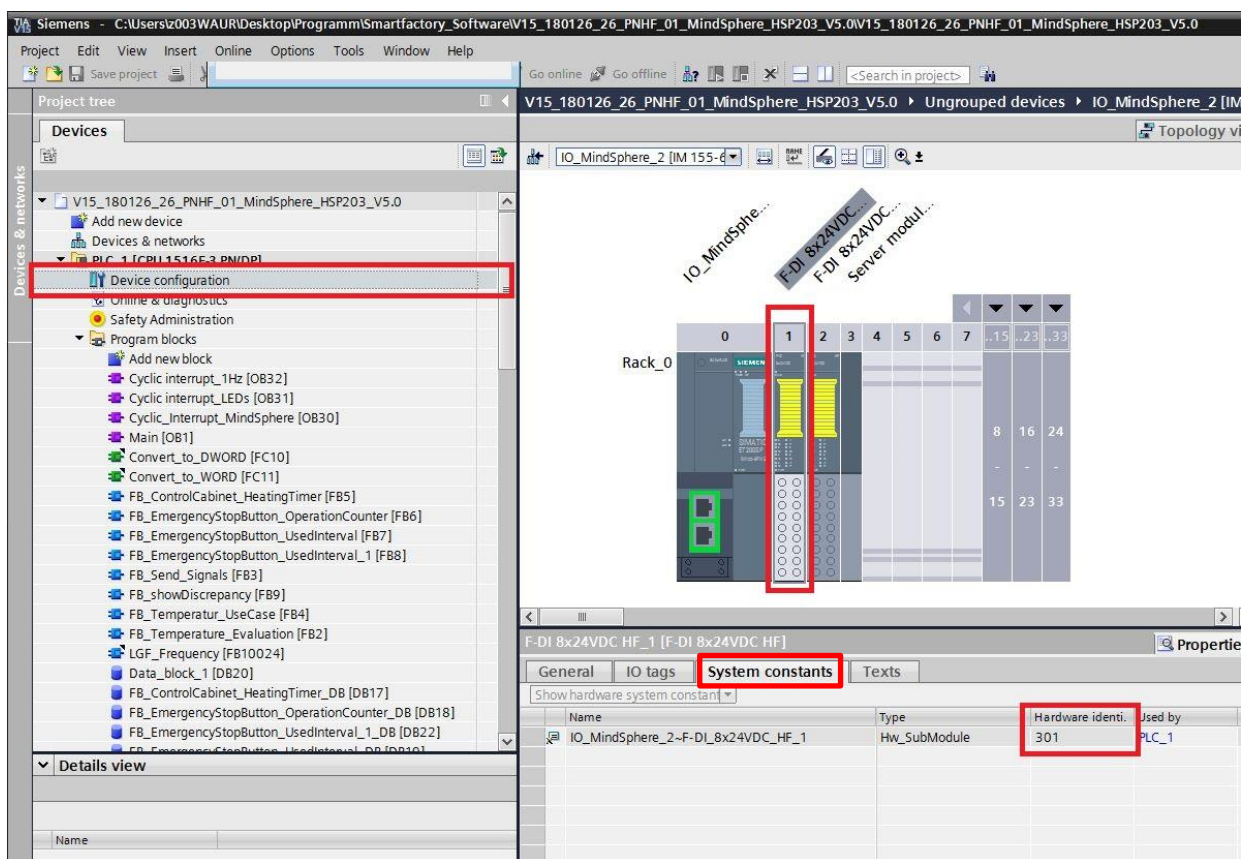


Abbildung 3.30: Hardware Nummer

Nun kann die erste MindSphere Source definiert werden. Dazu klickt man auf den Datenbaustein „MindSphere_Data0_Controll“ (Abbildung 3.31) und öffnet eine beliebige MindSphere Source. In diesem Beispiel, die erste Source mit dem Namen „MindSphereSource [0]“, welche in Abbildung 3.31 ersichtlich ist. Unter dem Punkt „HWID_Source“ muss nun die Nummer des gewünschten Moduls eingetragen werden. In diesem Fall die Nummer des F-DI Moduls aus der dezentralen Peripherie mit dem Namen MindSphere_2, wie oben beschrieben die Nummer 301. Bei „DSNumber“ muss die Nummer des Datensatzes eingetragen werden, in diesem Fall handelt es sich wie oben um ein F-DI

Modul, also der Datensatz 201. Unter „DSLengthRead“ versteht man noch die Länge des verwendeten Datensatzes, damit das TIA Portal weiß, wie viele Bit pro MindSphere Source reserviert werden müssen. Im letzten Punkt wird noch der Datentyp und somit auch der zuständige Datenbaustein definiert, an dem die Daten abgeholt werden. Mit der Eingabe des Wertes 101 im Feld „DataTypeDBNr“, bezieht sich das System auf den DB 101 mit dem Namen MindSphere_Data2_Word. In diesem befinden sich nun die Kontaktstellen zu MindSphere.

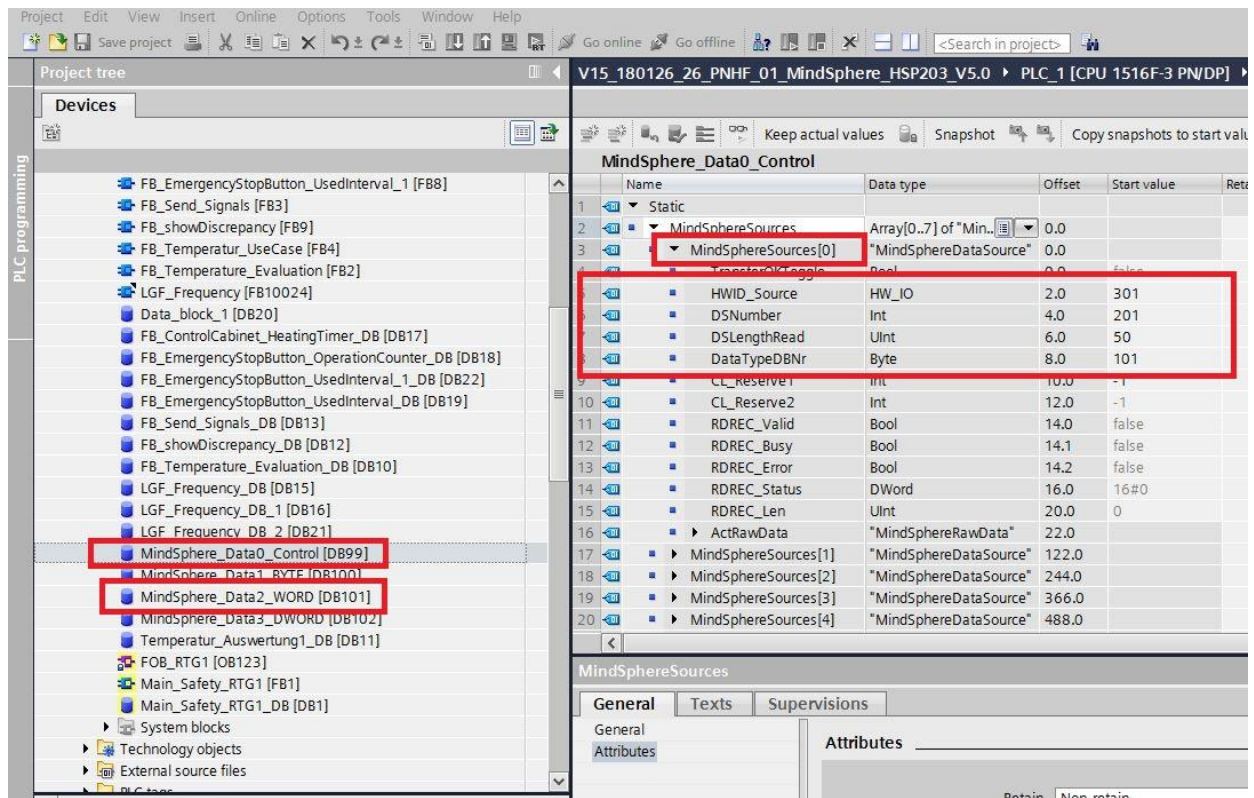


Abbildung 3.31: MindSphere Data Control

Klickt man nun auf den DB 101 und wählt wie zuvor die „Source 0“ aus, sieht man die einzelnen Elemente aufgelistet. Ersichtlich ist das in Abbildung 3.33.

Um zu verstehen welches Element mit welchem Bit verknüpft ist, wird einen Blick auf Abbildung 3.32 geworfen. Diese zeigt den Datensatz, der vom FSM ausgehen wird. Die ersten beiden Bits geben die CPU-Temperatur der Baugruppe wieder. Bit 18 und 19 zeigen die Diskrepanz in Prozent für den „inRange“ Fall an, 20 und 21 für den Fall „outRange“. Zuletzt zeigen noch die Bits 22 und 23 in welchem der beiden Fälle man sich befindet.

Datensatz 201 - F-DI															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Temperatur F-DI	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
reserved	reserved	Disprepanz in % für den In Range Fall	Disprepanz in % für den Out Range Fall	Disprepanz (In range = 1, Out Range = 2)	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved

Abbildung 3.32 Datensatz

Diese Daten aus den Datensätzen werden in die Elemente des DB101 geschrieben. Wie in Abbildung 3.33 ersichtlich, werden diese Bits nun Elementen zugewiesen. Für das Element [0] sind die ersten 2 Bits (0,1) reserviert, ersichtlich wird das bei der Betrachtung des Offsets. Das Element [1] bekommt die Bits 2 und 3. Mit diesem Wissen ergibt sich nun die Zuteilung der Datenbits. Das Element [0] gibt somit die Temperatur an Bit (0,1), das Element [9] Bit (18,19), Element [10] Bit (20,21) und Element [11] Bit (22,23) die Werte für die Diskrepanz. Mit diesen Elementen können nun die genauen Datenübergangspunkte definiert werden. Für die Temperatur zum Beispiel der Punkt aus dem DB101 mit dem Offset 0.0. Somit ist die Datenbereitstellung im TIA Portal abgeschlossen. Wie der Datenübergabepunkt in MindSphere konfiguriert wird, folgt im Kapitel 3.3 MindSphere Einrichtung.

The screenshot shows the Siemens TIA Portal interface. The Project tree on the left lists various data blocks, with 'MindSphere_Data2_WORD [DB101]' highlighted. The main window displays the configuration for 'MindSphere_Data2_WORD' as a Word data type. The table below shows the elements and their offsets:

Name	Data type	Offset	Start value
Static	Array[0..7] ...	0.0	
Sources[0]	"MindSphereData_..."	0.0	
Elements	Array[0..49] of Int	0.0	
Elements[0]	Int	0.0	0
Elements[1]	Int	2.0	0
Elements[2]	Int	4.0	0
Elements[3]	Int	6.0	0
Elements[4]	Int	8.0	0
Elements[5]	Int	10.0	0
Elements[6]	Int	12.0	0
Elements[7]	Int	14.0	0
Elements[8]	Int	16.0	0
Elements[9]	Int	18.0	0
Elements[10]	Int	20.0	0
Elements[11]	Int	22.0	0
Elements[12]	Int	24.0	0
Elements[13]	Int	26.0	0
Elements[14]	Int	28.0	0
Elements[15]	Int	30.0	0
Elements[16]	Int	32.0	0
Elements[17]	Int	34.0	0
Elements[18]	Int	36.0	0
Elements[19]	Int	38.0	0

Abbildung 3.33 DB101 MindSphere Word

3.3 MindSphere Einrichtung

In diesem Punkt werden die benötigten Hardware Komponenten sowie die dazugehörigen Programme beschrieben, um MindSphere zu konfigurieren, Apps auf die Plattform zu bringen und Datenübergabepunkte in MindSphere definieren zu können.

3.3.1 Vorinstallationen am Computer

In diesem Punkt sollen kurz die nötigen Schritte erläutert werden, um den verwendeten Computer für die MindSphere Einrichtung vorzubereiten. Mit Hilfe dieser Schritte kann MindSphere gestartet, Apps auf die Plattform geschickt und somit eine lokale Development Umgebung geschaffen werden. Die Punkte werden aufgezählt, um grob einen Überblick zu bekommen, um im Falle einer Neueinrichtung die Schritte einfach wiederholen zu können. Eine ausführliche Erklärung dieser Schritte findet im folgenden Kapitel statt.

1. Installation und Download von Cloud Foundry command shell (CF).
CF wird benötigt, um mit der SAP Cloud Foundry Plattform zu kommunizieren und Apps auf die Plattform zu spielen
2. Kompilieren des Quellcodes
Hierzu wird das Java Development Kit JDK benötigt, sowie auch Gradle 3.3. Diese Software wird zum Übersetzen des HTML Codes der App für Cloud Foundry benötigt. Eine detaillierte Anleitung für die Installation sowie die Links zum Download, ist im von Siemens bereitgestellten PDF „MindAccess Developer Tutorials“ zu finden.
3. Nach der Installation sollte der Login bei Cloud Foundry möglich sein, der genaue Ablauf ist in den Punkten unten beschrieben:
 - a. Öffnen der Windows Konsole, Eingabe des Befehls.
“cf login -a <https://api.cf.mindsphere.io-ss0>“
 - b. Danach wird eine URL ausgegeben, die im Browser geöffnet wird.
 - c. Im nächsten Schritt kann über die Konsole oder auch über den Browser der Platz für eine App zugewiesen werden.
4. Im letzten Schritt kann noch kontrolliert werden, ob eine App erfolgreich übertragen wurde:
 - a. Mit der Eingabe von cf -a, werden alle in der Cloud befindlichen Apps aufgelistet
 - b. Dies kann auch über den Browser mit der URL <https://login.cf.mindsphere.io/login> kontrolliert werden.

3.3.2 Hardware

Um die Daten von der SPS oder anderen Übergabepunkten in die Cloud zu senden, wird ein Gateway von Siemens verwendet. In dieser Arbeit wird das MindConnect IoT 2040 verwendet, daneben gibt es noch das MindConnect Nano für größere Datenmengen. In Abbildung 3.34 ist das IoT 2040 ersichtlich, vier Anschlüsse sind dabei von Bedeutung.



Abbildung 3.34: IoT 2040 [34]

Punkt 1 ist ein USB Anschluss zum Übertragen der Konfigurationsdateien, dessen Funktionsweise wird in 3.3.3 IoT Data Modeler beschrieben. Rechts im Bild zu sehen sind die LAN- Anschlüsse, der rechte stellt eine Verbindung zur SPS her und der linke Anschluss ermöglicht eine Verbindung mit dem Internet. In unserem Fall wird die rechte Buchse mit F1500 CPU verbunden und die linke mit einem LTE Router oder mit dem LAN Netzwerk der Universität.

In der nächsten Abbildung 3.35 ist noch der vierte Anschluss zu sehen. Dabei handelt es sich um die Stromversorgung der MindConnect, im Falle des Aufbaus wurde diese an das Stromnetz der SPS angeschlossen.

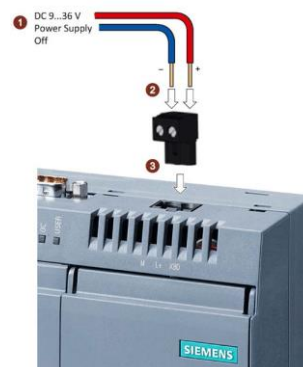


Abbildung 3.35: Stromanschluss [34]

Im nächsten Schritt wird die MindConnect konfiguriert. Beim Kauf von MindSphere erhält man einen URL-Link zu seinem MindSphere Account, in diesem Fall handelt es sich noch um eine Entwicklerversion. MindSphere lässt sich über diesen Link erreichen, zu Beginn

befinden sich nur Standard-Apps auf MindSphere, zusätzliche Apps können später mit den zuvor installierten Programmen gepusht werden. Zu den Standard-Apps gehören der IoT Data Modeler, der Fleet Manager, der Visual Analyzer und der Application Manager. In Abbildung 3.35 ist die Startoberfläche inklusive dieser Apps ersichtlich, auf welche in den nächsten Seiten genauer eingegangen wird.

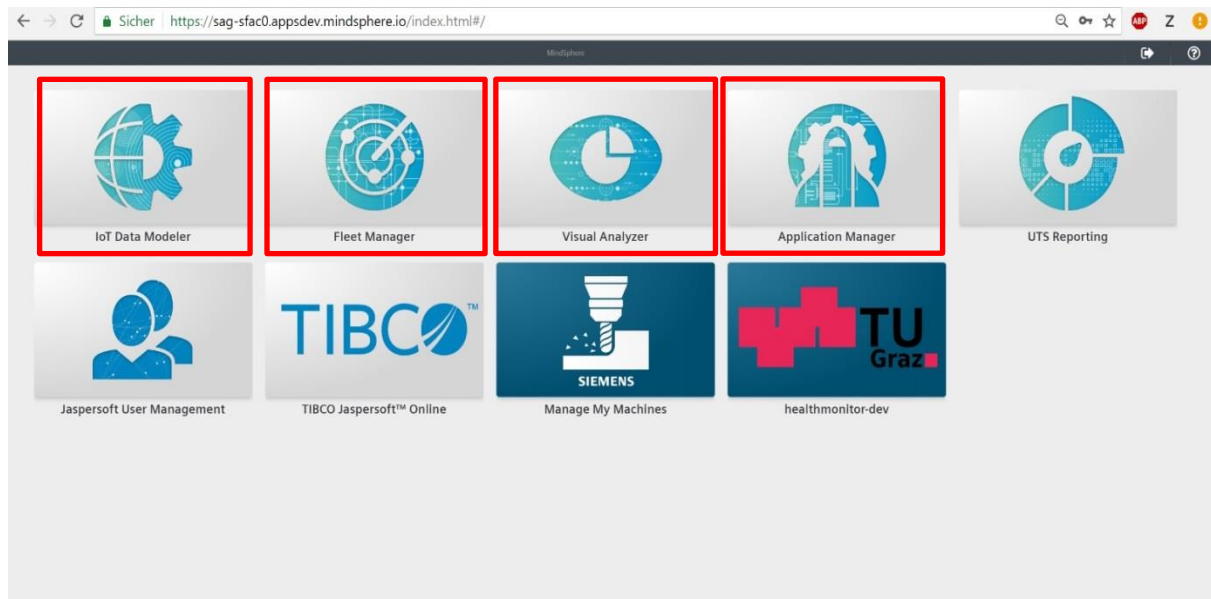


Abbildung 3.36: MindSphere Startseite mit Standard Apps (rote Kästchen)

3.3.3 IoT Data Modeler:

Klickt man auf die Data Modeler App, erscheint die Oberfläche, welche in Abbildung 3.37 ersichtlich ist. Sag-sfac0 ist dabei der von Siemens vergebene Name, Smart Factory ist das im Laufe der Arbeit hinzugefügte Asset, welches z.B. einer von mehreren Produktionsstandorten sein könnte. Im nächsten Schritt lässt sich ein Unterpunkt definieren, z.B. eine Maschine oder eine Steuerung, an welche Daten übergeben werden. Im Fall dieses Versuchsaufbaus handelt es sich um die SPS CPU F-1500. Ersichtlich ist die Oberfläche in Abbildung 3.37. Neue Maschinen oder Produktionsstandorte lassen sich einfach mit der Option „neues Asset hinzufügen“ erstellen.



Abbildung 3.37: Data Modeler

Danach erscheint die Oberfläche in Abbildung 3.38. Nach entsprechender Eingabe der Daten erscheint am Display die Auswahl (Abbildung 3.39) der verwendeten Hardware, im Falle dieser Arbeit handelt es sich um die MindConnect 2040.

Abbildung 3.38: Dateneingabefeld

Abbildung 3.39: Hardwareauswahl

Danach kann die MindConnect konfiguriert werden. Die Abbildung 3.40 bis Abbildung 3.43 zeigen die Konfiguration für die Smart Factory.

- Zuerst wird die Hardware ID-Nummer der MindConnect benötigt, diese ist auf die MindConnect aufgedruckt.

2. Konfiguration

Konfiguration

MindConnect IoT-2040 eindeutige Kennung

SV-J6B68964

Abbildung 3.40: Hardware ID Eingabe

- Im nächsten Schritt wird die IP-Adresse (Internetprotokoll) und die Subnetzmaske der angeschlossenen Hardware vergeben. 192.168.0.50 ist in unserem Fall die IP-Adresse der F1500 CPU, 255.255.255.0 die dazugehörige Subnetzmaske.

3. Netzwerk für Datenerfassung

Ethernet-Schnittstelle für die Verbindung zur Datenerfassung

Netzwerk-Verbindungsinformationen eingeben

DHCP lokales Netzwerk

IP-Adresse
192.168.0.50

Netzmaske
255.255.255.0




Abbildung 3.41: Schnittstellenkonfiguration Fabrik seitig

- Im nächsten Punkt werden die Verbindungsdaten der vorhandenen Internetverbindung zugewiesen. Im Fall eines LTE-Routers wird DHCP (Dynamic Host Configuration Protocol) verwendet. Mit diesem Protokoll wird automatisch eine IP-Adresse und Subnetzmaske hinzugefügt [35].

4. Netzwerk für MindSphere Verbindung

Ethernet-Schnittstelle für die Verbindung zur MindSphere

Netzwerk-Verbindungsinformationen eingeben

DHCP

IP-Adresse
IP-Adresse

Netzmaske
Netzmaske

Gateway
Gateway

DNS 1
DNS 1

DNS 2
DNS 2




Abbildung 3.42: Schnittstellenkonfiguration zur Cloud

Sollte die MindConnect an ein LAN-Netzwerk, wie in der Smart Factory (kein DHCP) angeschlossen werden, benötigt das Gateway eine statische IP Adresse und die zugehörige

Subnetzmaske. Diese muss vom IT-Beauftragten mit Hilfe der Mac-Adresse (Media-Access-Control-Adresse) zugewiesen werden. Diese Adressen müssen dann an dieser Stelle für die Einrichtung der MindConnect eingetragen werden.

- Bevor die Konfigurationsdatei nun per USB-Stick auf die MindConnect übertragen werden muss, kann noch die Einstellung für einen möglichen Proxy-Server, sowie die Einstellung für das Firmware Update gewählt werden. Proxy-Server sind in diesem Fall keine vorhanden und das Firmwareupdate wird manuell durchgeführt.
- Nun kann die konfigurierte Firmware auf den USB-Stick gespeichert werden. Beim Einspielen der Firmware traten immer wieder Probleme auf. Obwohl die richtigen Einstellungen gewählt wurden, war die MindConnect nicht in der Lage, online zu gehen. Da kein Fehler, auch nach Rücksprache mit dem Siemens Supportteam gefunden wurde, wurde der Konfigurationsablauf mehrfach wiederholt. Auch ein File zum Auslesen des Fehlerspeichers der MindConnect brachte keine Fehler zum Vorschein. Einziger Lösungsweg welcher zum Ziel führte, war das wiederholte Aufspielen der Konfigurationsdatei. Leider gibt es für dieses Problem keinen besseren Lösungsvorschlag. Das Aufspielen der zuvor erstellten Konfigurationsdatei hat funktioniert, wenn unter Agent der Punkt „Onboarded“ grün erscheint und drei grüne Lampen auf der MindConnect leuchten.

Onboarded	Online	Name	Type
● Onboarded	●	SV-J6868964	pico.mindconnect.pico

Abbildung 3.43: Online Status

Um die Daten von der SPS an MindSphere übergeben zu können, müssen nun noch Datenpunkte definiert werden, dies geschieht auch im IoT Daten Modeler.

Auf der MindSphere Startseite klickt man hierzu wieder auf die App IoT Daten Modeler. Klickt man nun auf Aspects und Add Aspect erscheint danach die Oberfläche, welche in Abbildung 3.44 ersichtlich ist. Bevor die genauen Datenübergangspunkte definiert werden, muss ein Aspect konfiguriert werden. Als erstes wird der Aspect Name definiert, im Fall des Aufbaus „FailSafeModules“. Unter dem Punkt „Data Source Protocol“ wird das verwendete Übertragungsprotokoll ausgewählt. Zur Auswahl stehen dabei OPC-UA und S7. OPC-UA ist (Open Platform Communications Unified Architecture) ein herstellerunabhängiges Kommunikationsprotokoll für M2M Anwendungen, sowie für alle anderen Anwendungen in der Digitalisierung, um die Kommunikation von Komponenten unterschiedlicher Hersteller zu ermöglichen [36]. Bei S7 handelt es um das interne Siemens-Protokoll zur Kommunikation zwischen Siemens Komponenten wie einer SPS [1].

Danach kann im Punkt „Reading Cycle“ noch die Abtastrate eingestellt werden, der kleinste mögliche Wert ist dabei eine Sekunde.

Details Agent Aspects

Aspect Configuration:

Aspect Name: FailSafeModules Data Source Protocol: S7

Reading Cycle: 1 s Active: Yes

[Add Variable](#)

Name	Unit	Address	Data Type	On data change (ODC)
FDI0TempCPU	°C	DB101.DBW0	sINT	<input type="checkbox"/>
FDI2TempCPU	°C	DB101.DBW200	sINT	<input type="checkbox"/>
FDI3TempCPU	°C	DB101.DBW300	sINT	<input type="checkbox"/>
FDI4TempCPU	°C	DB101.DBW400	sINT	<input type="checkbox"/>
FDI1Discrepancy		DB101.DBW118	sINT	<input type="checkbox"/>
FDI1DiscrepancyTyp		DB101.DBW122	sINT	<input type="checkbox"/>
FDI1DiscrepancyOn		DB101.DBW120	sINT	<input type="checkbox"/>
TimeDifff	t	DB22.DB02	sINT	<input type="checkbox"/>
countNumber		DB18.DBW0	sINT	<input type="checkbox"/>

Configuration:

IP Address of S7: 192.168.0.1

Abbildung 3.44: Aspects Konfiguration

Sind diese Schritte erledigt, können nun die echten Datenpunkte eingegeben werden. Dazu fügt man unter Add Variable eine neue Variable hinzu.

Wichtig ist dabei die Eingabe des Namens, da dieser später von den verschiedenen Apps weiterverwendet wird. **Die Apps greifen also auf den Namen der Variable zu und nicht auf die danebenstehende Adresse. Die verwendete Adresse gibt nur an, wo der Datenpunkt in der SPS liegt.** Wie im Kapitel davor erwähnt, werden dazu die Datenbausteine im TIA Portal modifiziert, um MindSphere die Möglichkeit zu geben, die Daten auszulesen.

Als Beispiel wird die Temperatur FDI2TempCPU erklärt. FDI steht dabei wieder für das digitale Input Modul, die Nummer 2 gibt an, wo sich das Modul befindet, in diesem Fall also am zweiten Steckplatz und TempCPU beschreibt, dass die Temperatur des Moduls ausgelesen wird. Den Wert erhält MindSphere aus der Adresse **DB101.DBW200**. DB steht für Datenbaustein DBW für den Datentyp der übertragen werden soll. Die unterschiedlichen Typen sind unten in der Tabelle 2 ersichtlich.

Data bit	DBX	1 Bit	Bool
Data word	DBW	2 Byte	Int
Data double word	DBD	4 Byte	Real

Tabelle 2: Datentypen

Daraus kann man die verschiedenen Datentypen und die benötigten Speicherplätze ablesen. Die benötigten Speicherplätze wurden zuvor schon im TIA Portal zugewiesen. Um die richtige Adresse zu erhalten, sollte im TIA Portal der gewünschte Datenbaustein aufgerufen werden. In diesem Baustein steht der Datentyp, die Nummer des Datenbausteins und der Offset, der die zweite Zahl angibt. In Abbildung 3.45 ist also ersichtlich, dass die „FDI2TempCPU“ aus dem „MindSphere_Data_WORD“ Datenbaustein mit der Nummer **101** besteht und einen Offset von **200** aufweist. Mit diesen Angaben ist es für MindSphere möglich, den Wert „**DB101.DBW200**“ auszulesen. Zuvor wurde beschrieben, dass das „Element[0]“ einer F-DI laut des bereitgestellten Datensatzes die Temperaturwerte der CPU enthält. Somit bleiben für das „Element[0]“ mit dem Offset „200.0“ zwei Bits für den Temperaturwert frei. Durch das bereits zuvor erklärte Eintragen der Hardware ID weist der Data Modeler nun genau von welchem Modul er die Temperatur Werte bekommt.

4	Sources[1]	"MindSphereData_...	100.0	
5	Sources[2]	"MindSphereData_...	200.0	
6	Elements	Array[0..49] of ...	200.0	
7	Elements[0]	Int	200.0	0
8	Elements[1]	Int	202.0	0
9	Elements[2]	Int	204.0	0
10	Elements[3]	Int	206.0	0
11	Elements[4]	Int	208.0	0
12	Elements[5]	Int	210.0	0
13	Elements[6]	Int	212.0	0
14	Elements[7]	Int	214.0	0
15	Elements[8]	Int	216.0	0
16	Elements[9]	Int	218.0	0
17	Elements[10]	Int	220.0	0

Abbildung 3.45: MindSphere Datenbaustein

3.3.4 Apps

Im nächsten Punkt werden die in dieser Arbeit verwendeten Apps beschrieben. Zu unterscheiden ist dabei zwischen vorinstallierten Apps, die sich auf jeder MindSphere befinden (siehe Abbildung 3.36) und die Apps für Temperatur und Diskrepanzüberwachung, welche von Siemens Graz selbst entwickelt wurde. Diese selbst entwickelten Apps müssen speziell mit Hilfe der Eingabe Konsole von Windows auf MindSphere gepusht werden. Eine detaillierte Erklärung ist später in diesem Kapitel zu finden.

Öffnet man MindSphere im Browser und gibt Adresse von MindSphere (<https://sag-sfac0.appsdev.mindsphere.io>) ein, erhält man folgende Ansicht (Abbildung 3.46):

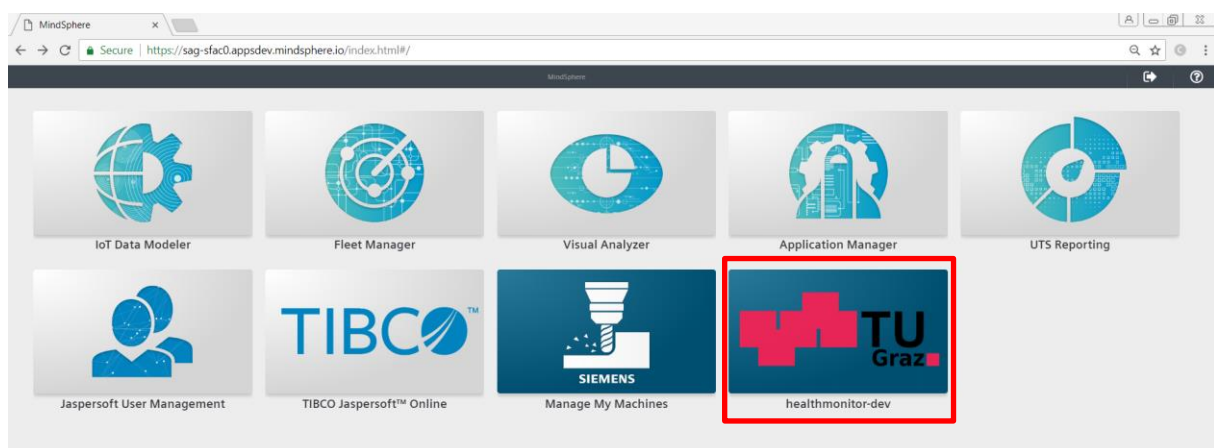


Abbildung 3.46: Startseite von MindSphere

Darauf erkennt man verschiedene Apps, wie der zuvor beschriebene IoT Daten Modeler, oder die von Siemens in Graz entwickelt App healthmonitor.

3.3.4.1 Standard-Apps

Im Fleet Manager als auch im Visual Analyzer können die Daten aufbereitet und in Diagrammen dargestellt werden. Der Visual Analyzer bietet gegenüber dem Fleet Manager die Möglichkeit, nur gewisse Datenpunkte darzustellen und in Graphen zu verarbeiten. Diese App muss jedoch zusätzlich erworben werden, der Fleet Manager kann hingegen kostenlos in MindSphere verwendet werden. Da sich diese Apps sehr ähneln und nur um Funktionen erweitert wurden, wird in diesem Punkt der Visual Analyzer behandelt. Der Visual Analyzer bietet den Vorteil zwischen mehr Darstellungsoptionen wählen zu können - es lassen sich verschiedene Diagramme auswählen, die individuell gestaltet werden können.

In Abbildung 3.47 ist die Visual Analyzer App ersichtlich. Der orange markierte Bereich zeigt die Maschine oder die Hardware, an der die Daten erfasst werden. In diesem Fall die SPS-CPU mit dem Namen CPU1516F. Im roten Kasten kann gewählt werden, welche Werte angezeigt werden sollen. Dazu kann im blauen Kasten (Abbildung 3.48) genau konfiguriert

werden, welche Werte dargestellt werden sollen, im Fall der Abbildung 3.47 wurden die Temperaturen ausgewählt.

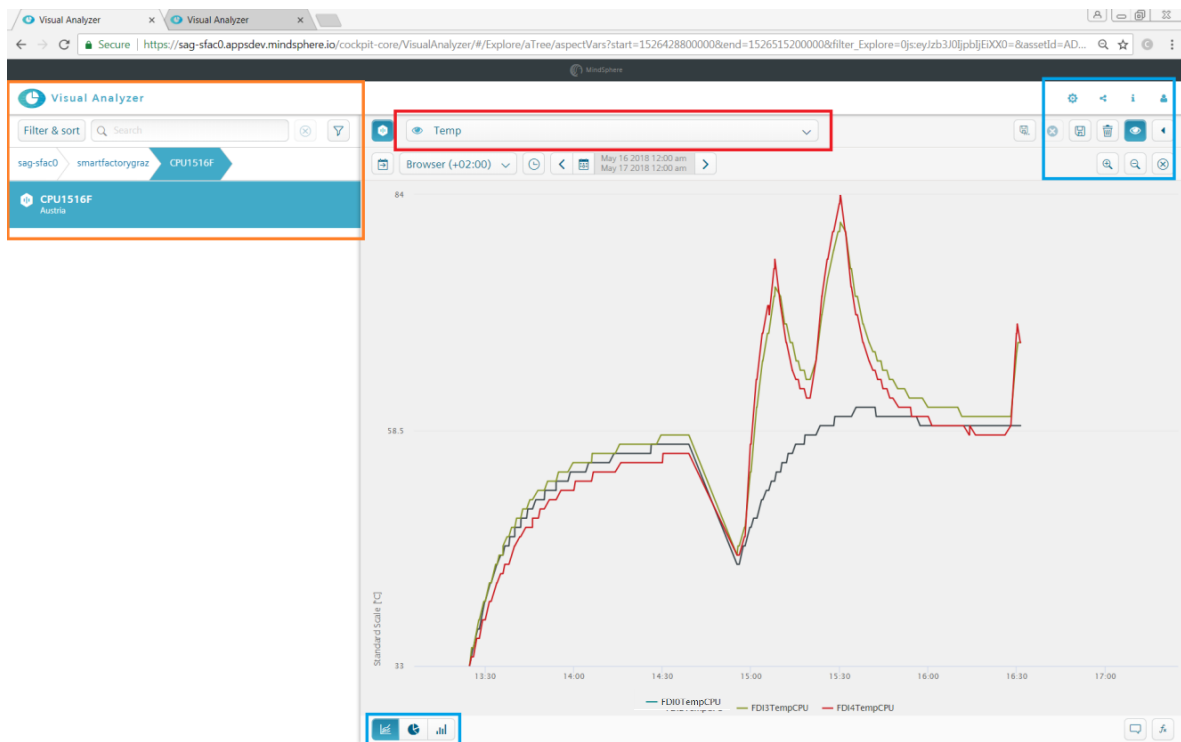


Abbildung 3.47: Temperaturanzeige

Ersichtlich ist der Vorgang in Abbildung 3.48, der Vorgang zu Erstellung ist selbsterklärend.

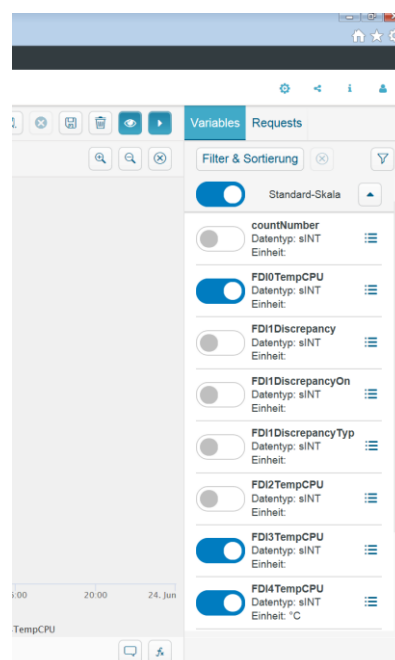


Abbildung 3.48: Auswahl der gewünschten Daten

Für die Arbeit wurden somit folgende Verläufe dargestellt:

- Temperatur
- Anzahl der Auslösungen des Not-Aus-Tasters,
- Diskrepanzzeiten,
- online Status der MindConnect
- Benutzungsintervall des Not-Aus-Tasters.

dargestellt. Ersichtlich sind die Einstellmöglichkeiten in Abbildung 3.48. Zusätzlich besteht noch die Option, die Daten in Balken- oder Tortendiagrammen darstellen zu lassen. Weiter soll in dieser Arbeit nicht auf die App eingegangen werden, da Siemens Tutorials und Leitfäden zur Benutzung zur Verfügung stellt.

3.3.4.2 Healthmonitor, Push App

3.3.4.2.1 Push App

Der Healthmonitor ist eine von Siemens in Graz entwickelte App zur Darstellung der Temperaturen von FSM. Zusätzlich kann noch die Diskrepanz von Sensorik ermittelt werden. Bevor die App benutzt wird, muss diese auf MindSphere installiert bzw gepusht werden. In den nächsten Punkten wird das Pushen ebenfalls erklärt, falls die App aktualisiert werden soll.

Für diese Installation wird nebst einem Computer noch ein Mobiltelefon benötigt. Auf diesem muss der SAP Authenticator installiert werden. Diese App ist ein zusätzlicher Sicherheitsmechanismus, der Einmalpasswörter generiert, um grundlegende Änderungen an MindSphere vorzunehmen. Das am Smartphone kreierte Passwort muss anschließend im Browser eingegeben werden. Bevor die App gepusht wird, muss der Computer dafür vorbereitet werden, dazu müssen wie zuerst oben beschrieben folgende Programm installiert werden:

- Cloud Foundry command shell
- JDK 1.8
- Gradle 3.3

Diese Programme dienen dazu, den verwendeten Code für die Cloud-Plattform zu kompilieren und diesen mit Hilfe der Konsole auf MindSphere zu bringen. Die genaue Installationsanleitung der einzelnen Programme wird ausführlich in einem öffentlich zugänglichen Paper von Siemens erklärt. Zu finden ist das PDF unter dem Namen „Mind_Access_Developer_Tutorials“.

Nach erfolgreicher Installation der Tools kann das Pushen der App beginnen. Dazu öffnet man im ersten Schritt die Windows Konsole.

Sollte ein Proxy Server vorhanden sein, so muss dieser zuerst gesetzt werden. Die Daten dafür erhält man beim IT-Dienst. Dies ist meist der Fall, wenn man sich im Netzwerk einer Firma oder der Universität befindet. Dazu wird der Befehl SET eingegeben und danach die URL des Servers.

- SET http_proxy=http://coia.hcvpc.io:9400/
- SET https_proxy=https://coia.hcvpc.io:9400/

Ersichtlich ist dieser Vorgang in Abbildung 3.49, in den Zeilen über dem blau markierten Bereich.

Im nächsten Schritt muss man sich bei CloudFoundry einloggen. Dazu wird, wie in Abbildung 3.49 ersichtlich, der Befehl „cf login -a https://api.cf.mindsphere.io -sso“ eingegeben.

```

C:\WINDOWS\system32\cmd.exe
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\z003WAUR>SET http_proxy=http://coia.hcvpc.io:9400/

C:\Users\z003WAUR>SET https_proxy=https://coia.hcvpc.io:9400/

C:\Users\z003WAUR>cf login -a https://api.cf.mindsphere.io -sso
API endpoint: https://api.cf.mindsphere.io

One Time Code ( Get one at https://login.cf.mindsphere.io/passcode )>
Authenticating...
  
```

Abbildung 3.49: Screenshot der Konsole, Login und Passwort (passcode)

In der Folge muss ein Passwort eingegeben werden. Dazu wird die in Abbildung 3.49 vor dem roten Kasten befindliche URL aufgerufen. Im Browser erscheint Abbildung 3.50.

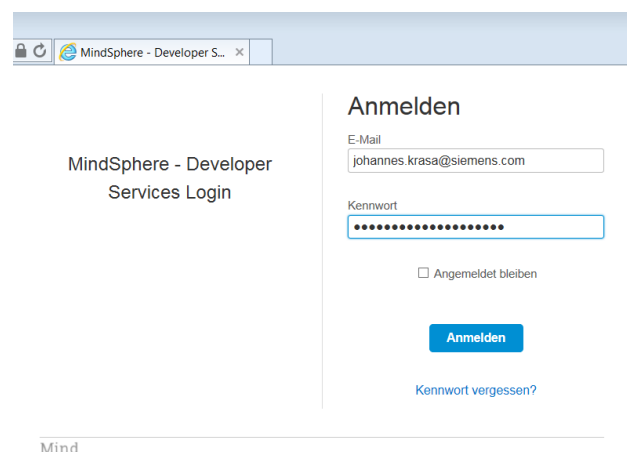


Abbildung 3.50: Passwortgenerierung im Browser

Nach Eingabe der Benutzerdaten und des Passworts (smartFactory@TU_GrazIFT) kommt man zur Zwei-Faktor-Authentifizierung (Abbildung 3.51). Als Sicherheitscode muss der in der Handy App (Abbildung 3.52) generierte Code verwendet werden. Dieser wird nach jeder Verwendung neu generiert.

Zwei-Faktor-Authentifizierung

Geben Sie zum Fortfahren den von Ihrem mobilen Endgerät generierten Sicherheitscode ein

E-Mail

Sicherheitscode*

*Eingabe erforderlich

Weiter

Mind
your digital future

Abbildung 3.51: Zwei-Faktor-Authentifizierung

Hierbei muss man sich zuvor mit seiner E-Mail-Adresse bei der App anmelden.

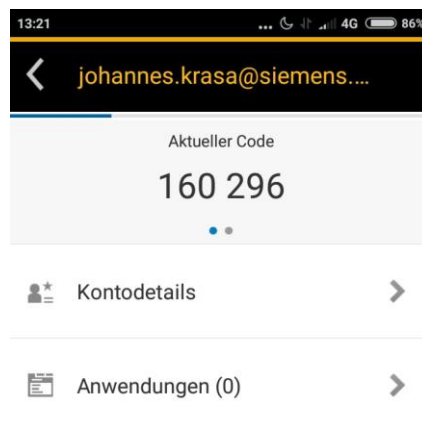


Abbildung 3.52: Screenshot der Handy App

Nach Eingabe der Zahlenkombination im Browser erscheint eine Seite (Abbildung 3.53) mit einem Passwort für die Konsole, welches in das rote Kästchen aus Abbildung 3.49 eingefügt werden muss.

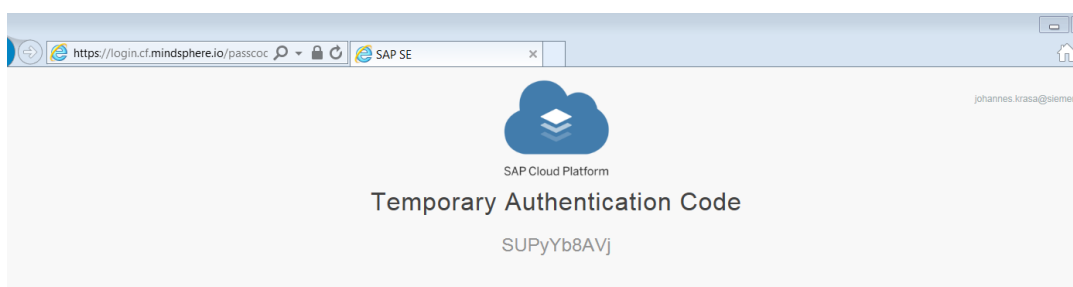


Abbildung 3.53: Authentication Code

Das Passwort lässt sich am einfachsten in die Konsole kopieren, das eingegebene Passwort ist aber **nicht ersichtlich**. Ist der Login erfolgreich, erscheint das grüne Kästchen aus Abbildung 3.54, welches die Userdaten sowie den von Siemens zugewiesenen Namen angibt.

```

C:\WINDOWS\system32\cmd.exe
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\z003WAUR>SET http_proxy=http://coia.hcvpc.io:9400/
C:\Users\z003WAUR>SET https_proxy=https://coia.hcvpc.io:9400/
C:\Users\z003WAUR>cf login -a https://api.cf.mindsphere.io -sso
API endpoint: https://api.cf.mindsphere.io

One Time Code ( Get one at https://login.cf.mindsphere.io/passcode )>
Authenticating...
OK

Targeted org sag-sfac0

Targeted space smartfactory

API endpoint: https://api.cf.mindsphere.io (API version: 2.106.0)
User:      johannes.krasa@siemens.com
Org:      sag-sfac0
Space:    smartfactory

C:\Users\z003WAUR>

```

Abbildung 3.54: Screenshot der Konsole

Nun muss die Konsole wissen wo die App liegt, dazu wird der Befehl „cd“ verwendet um den Zugriff auf den App Ordner zu leiten. Wie in Abbildung 3.55 ersichtlich, lag die App beim Aufspielen auf dem Desktop im Ordner Smart Factory. Am einfachsten erhält man den Link indem die URL aus dem Explorer kopiert.

```

C:\WINDOWS\system32\cmd.exe
C:\Users\z003WAUR\Desktop\SmartFactory_Code\signaling-app>dir
Volume in drive C is SYSTEM
Volume Serial Number is 68A3-74DA

Directory of C:\Users\z003WAUR\Desktop\SmartFactory_Code\signaling-app

06.03.2018  14:00    <DIR>          .
06.03.2018  14:00    <DIR>          ..
06.03.2018  12:15             31 .cfignore
06.03.2018  12:15             20 .gitignore
06.03.2018  14:00    <DIR>          app
06.03.2018  12:15            284 app.js
06.03.2018  14:00    <DIR>          css
06.03.2018  12:15           1.150 favicon.ico
06.03.2018  14:00    <DIR>          images
06.03.2018  12:15           6.919 index.html
06.03.2018  14:22            216 manifest.yml
06.03.2018  12:15            727 package.json
06.03.2018  12:15             38 readme.md
06.03.2018  14:00    <DIR>          utils
06.03.2018  14:00    <DIR>          views
                8 File(s)          9.385 bytes
                7 Dir(s)  54.581.739.520 bytes free

C:\Users\z003WAUR\Desktop\SmartFactory_Code\signaling-app>

```

Abbildung 3.55: Eingabe des Pfades

In die Konsole wird hierzu „cd“ und der angegebene Link aus dem Explorer eingegeben. Danach erhält man den richtigen Pfad.

Im letzten Schritt wird die App mit dem Befehl (Abbildung 3.56) „cf push“ in die Cloud übertragen. Wurde dieser Befehl erfolgreich ausgeführt, erhält man ein OK in der Konsole und die App befindet sich danach in der Cloud. Überprüft werden kann dies wie zuvor beschrieben mit dem Befehl „cf -a“. Nach Eingabe erscheint eine Liste mit allen auf MindSphere befindlichen Apps.

```
C:\Users\z003WAUR\Desktop\SmartFactory_Code\signaling-app>cf push
Pushing from manifest to org sag-sfac0 / space smartfactory as johannes.krasa@siemens.com...
Using manifest file C:\Users\z003WAUR\Desktop\SmartFactory_Code\signaling-app\manifest.yml

Deprecation warning: Route component attributes 'domain', 'domains', 'host', 'hosts' and 'no-hostname' are deprecated. Found: domain, host.
Please see http://docs.cloudfoundry.org/deuguide/deploy-apps/manifest.html#deprecated for the currently supported syntax and other app manifest deprecations. This feature will be removed in the future.

Using manifest file C:\Users\z003WAUR\Desktop\SmartFactory_Code\signaling-app\manifest.yml

Creating app healthmonitor-dev in org sag-sfac0 / space smartfactory as johannes.krasa@siemens.com...
OK

Creating route healthmonitor-dev-sag-sfac0.appsdev.mindsphere.io...
OK

Binding healthmonitor-dev-sag-sfac0.appsdev.mindsphere.io to healthmonitor-dev...
OK

Uploading healthmonitor-dev...
Uploading app files from: C:\Users\z003WAUR\Desktop\SmartFactory_Code\signaling-app
```

Abbildung 3.56: App pushen

Um nun die App in MindSphere nutzen zu können, muss diese noch mit dem Introduction Application Manager konfiguriert werden, damit sie in MindSphere sichtbar wird. Dazu öffnet man im Browser die MindSphere Startseite und öffnet den Introduction Application Manager, wie in Abbildung 3.57 ersichtlich.

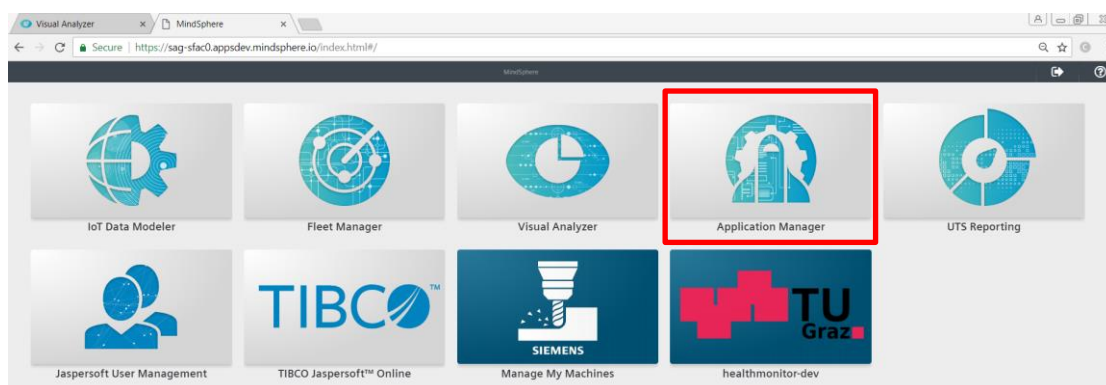


Abbildung 3.57: Introduction Application Manager

Diese Schritte sind notwendig, um die App im Startmenü erscheinen zu lassen. Dazu muss die App registriert werden. Dafür ist ein Name, die URL zum Aufrufen sowie ein Bild, mit dem die App später auf dem Startbildschirm erscheint, zu vergeben. Sind diese Punkte erfüllt und wurde der Lizenzvereinbarung zugestimmt, ist der Ablauf abgeschlossen und die App befindet sich in MindSphere. Ersichtlich ist der Vorgang in Abbildung 3.58.

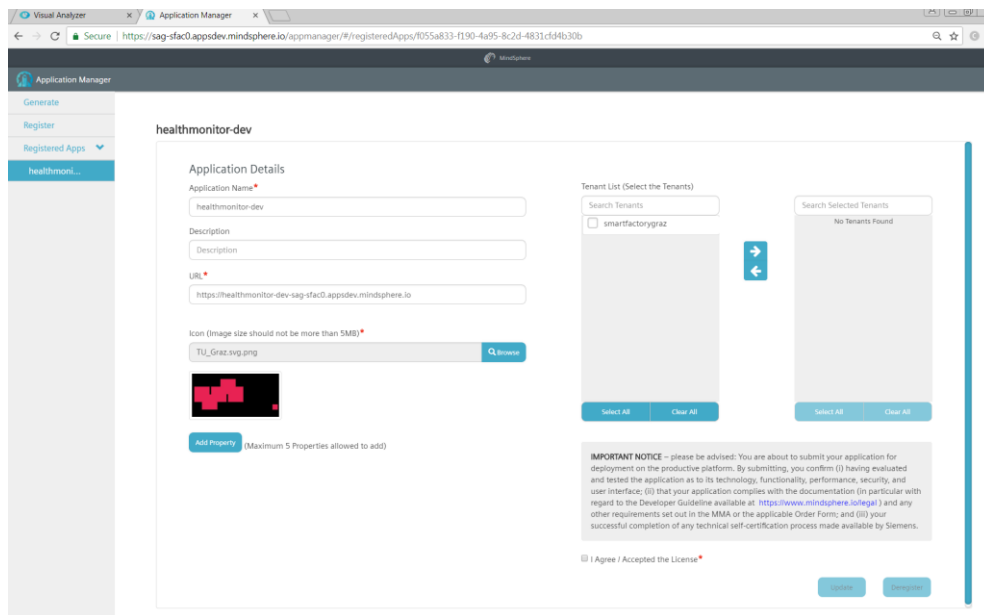


Abbildung 3.58: Eingabe der Daten im Application Manager

3.3.4.2.2 Healthmonitor

In diesem Punkt wird der Aufbau und die Funktion kurz erklärt sowie die Änderungen, die für diese MindSphere Plattform vorgenommen wurden.

Klickt man nun auf die App erscheint diese wie in Abbildung 3.59. Unter „Select Source“ kann ausgewählt werden, ob die Temperatur oder der Diskrepanz Fall angezeigt wird.

Die App selber ist mit Ampeln aufgebaut, welche die Temperatur anzeigen, die Schwellwerte sind dabei in unserem Fall gerade 55 und 65 Grad Celsius. Steigt die Temperatur in einem FSM, so ändert sich die Farbe. Somit können Unregelmäßigkeiten in einzelnen Modulen erfasst werden.

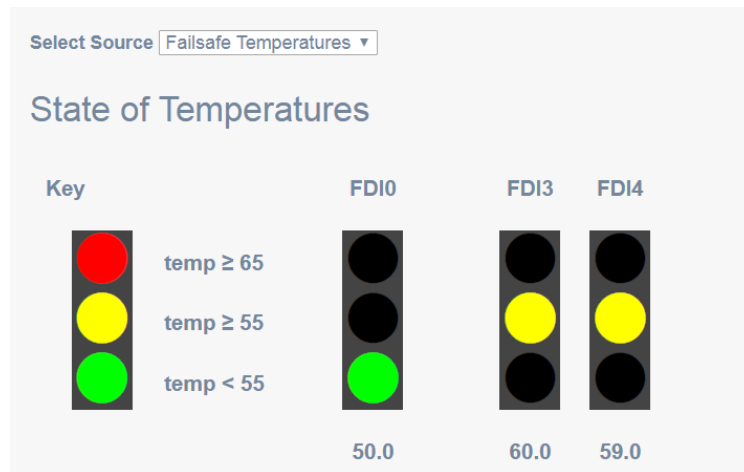


Abbildung 3.59: Healthmonitor App

Die zweite Funktion der App ist die Messung der Diskrepanz zweier Sensoren, dazu wird in der Firmware der Module eine Zeitspanne definiert, die als Reaktionszeit nicht überschritten werden darf.

Die App zeigt an, ob das Signal „in range“ oder „out of range“ ist und in welchem Prozentrahmen sich die Abweichung befindet. Beim „in range“ Fall ist der Balken grün, wird dieser rot, wurde die zulässige Zeit überschritten und die App wird einen „out of range“ Fall darstellen. Auch dieses etwaige Überschreiten wird wieder in Prozent der überschrittenen Zeit angegeben.

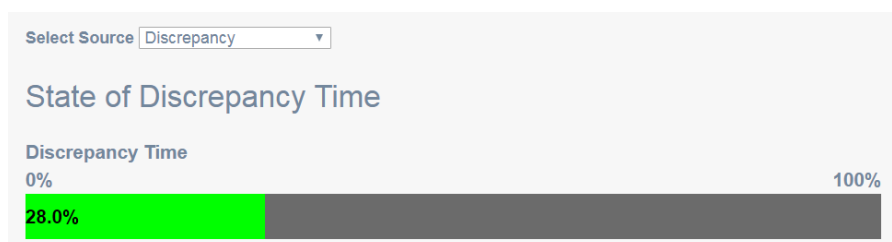


Abbildung 3.60: Discrepancy Messung

Anpassungen an der App, um diese für andere MindSphere Tenants nutzen zu können:

In diesem Punkt soll nun geklärt werden, wie die App modifiziert wird, um für den jeweiligen MindSphere Tenant die richtigen Einstellungen zu wählen. Unter Tenant ist der von Siemens zu Verfügung gestellte Speicherplatz in der Cloud zu verstehen. Jeder MindSphere Kunde bekommt im Zuge der Zuweisung, pro Tenant eine URL. Sprich jeder Tenant verfügt über ein eigenes MindSphere System mit eigener URL. Jeder dieser Tenants ist über den Namen zu erkennen. Wird die App nun auf einen anderen Tenant gespielt muss dieser in der App neu

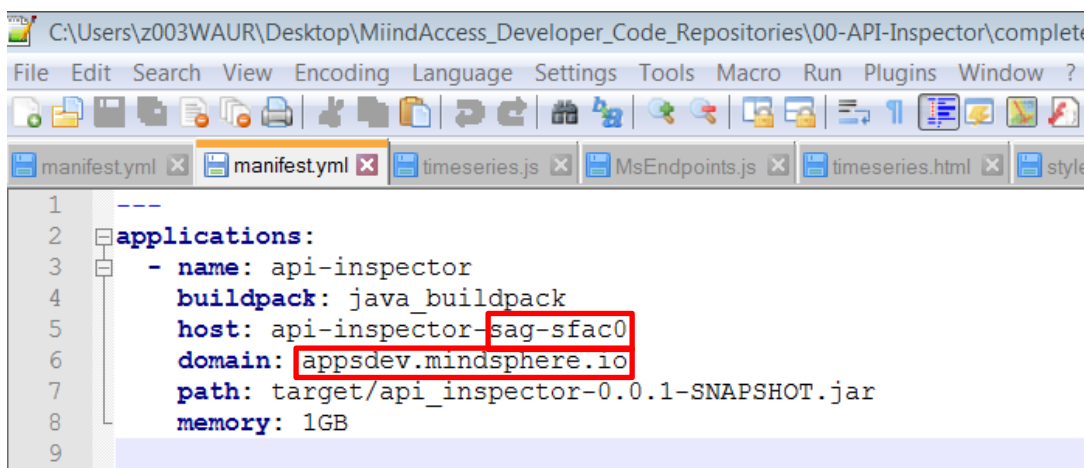
eingetragen werden. Zusätzlich soll erklärt werden, wie die App geändert wird, falls es zu einer Änderung der Datenübertragungspunkte im IoT Modeler kommt oder der Name bzw. der Tenant geändert wird.

Zuerst muss das Manifest der App geändert werden. Unter einem Manifest versteht man in der App-Entwicklung eine Datei, die alle Informationen über eine App bereitstellt [37]. Darunter befinden sich zum Beispiel der Name, der Autor, das Icon. Mithilfe des Manifests kann MindSphere beim Pushen einfach alle Daten aus dem Manifest auslesen und so die App in den richtigen Tenant pushen. Dazu öffnet man den Ordner, in dem die App liegt und verwendet für das Ändern einen Text Editor wie zum Beispiel das Notepad. Öffnet man das Manifest, erhält man die Ansicht aus Abbildung 3.61. Die dort vorgenommenen Einstellungen beziehen sich auf den derzeitigen MindSphere Tenant mit dem Namen `https://sag-sfac0.appsdev.mindsphere.io`.

Wichtig dabei ist, dass unter dem Punkt „host“ und „domain“ im Falle einer Änderung der MindSphere Adresse die letzten beiden Buchstabenkombinationen geändert werden. Diese beziehen sich nämlich auf die URL der vorhandenen MindSphere Adresse.

Erwähnt wird es hier, da zum jetzigen Zeitpunkt ein Entwickler Account genutzt wird. Es ist durchaus möglich, dass der Account umgestellt wird oder die App auf einen anderen MindSphere Account gepusht werden muss. Die gleiche Änderung muss auch vollzogen werden, wenn eine neue Version von MindSphere verwendet werden soll.

Die neue Domain wird vom MindSphere Support bekannt gegeben.



```

1  ---
2  applications:
3  - name: api-inspector
4    buildpack: java_buildpack
5    host: api-inspector-sag-sfac0
6    domain: appsdev.mindsphere.io
7    path: target/api_inspector-0.0.1-SNAPSHOT.jar
8    memory: 1GB
9

```

Abbildung 3.61 Manifest

Sind nun der Host und die Domain richtig definiert, kann mit der Anpassung der App begonnen werden. Dazu müssen Daten aus dem IoT Modeler ausgelesen werden. Unter den Aspects befinden sich die zuvor definierten Übergangspunkte. Für die Temperatursauswertung werden die Temperaturen der Module FDI0, FDI3 und FDI4 verwendet. Die Diskrepanzwerte der Endlagensensorik werden über das FDI1 ausgelesen. Die Zahlen geben wieder die Anordnung der Module an. Das Modul 0 befindet sich dabei in

der dezentralen Peripherieeinheit (MindSphere_2) unter dem Tisch, die Module 3 und 4 im Schaltschrank (MindSphere_1) für das Temperatur-Case. Dadurch lässt sich ein Vergleich zwischen den Temperaturen im Schaltschrank und den Temperaturen außerhalb ziehen.

The screenshot shows the 'Aspects' tab in the IoT Modeller. Under 'Aspect Configuration', the 'Aspect Name' is 'FailSafeModules', 'Reading Cycle' is '1 s', 'Data Source Protocol' is 'S7', and 'Active' is 'Yes'. Below this is a table of 'Variables (9)' with columns for Name, Unit, Address, Data Type, and On data change (ODC).

Name	Unit	Address	Data Type	On data change (ODC)
FDI0TempCPU	°C	DB101.DBW0	siINT	
FDI2TempCPU	°C	DB101.DBW200	siINT	
FDI3TempCPU	°C	DB101.DBW300	siINT	
FDI4TempCPU	°C	DB101.DBW400	siINT	
FDI1Discrepancy		DB101.DBW118	siINT	
FDI1DiscrepancyTyp		DB101.DBW122	siINT	
FDI1DiscrepancyOn		DB101.DBW120	siINT	
TimeDiff	t	DB22.DBD2	siINT	
countNumber		DB18.DBW0	siINT	

Abbildung 3.62: IoT Modeller Datenpunkte und Namen

Nun muss die App noch wissen, welche Daten sie abrufen kann. Dazu muss diese so modifiziert werden, dass diese auf die richtigen Werte zugreifen kann. Dazu brauchen man die „assetId“, diese kommen aus dem IoT Modeller indem man auf das gewünschte Asset klickt und die letzte Zahlenfolge vor dem Gleichheitszeichen ausliest. In unserem Fall erhält man die Asset ID der CPU1516F.

The screenshot shows the 'Asset Details' page for 'CPU1516F'. The browser address bar shows the URL: <https://sag-pro0.appsdev.mindsphere.io/mindspheremanager/9B89D3539FF546B3BC5201B355ADCE51/details?selected=6819A1EB7C8E406AB5AC30DAB1E4BE6D>. The 'Asset Identifier' field is highlighted with a red box.

Asset Details	
Name	CPU1516F
Asset type	
Asset Identifier	6819A1EB7C8E406AB5AC30DAB1E4BE6D

Abbildung 3.63: Asset ID

Diese „assetId“ muss nun in die App geschrieben werden. Dazu öffnet man im App-Ordner das File „timeseries.js“ mit dem Notepad (Abbildung 3.64). In dieser Datei muss die „self.assetId“ angepasst werden. In Abbildung 3.64 erkennt man die grün hinterlegten auskommentierten „assetIds“, diese ID sind für den Versuch andere Assets an die App zu

schließen benötigt worden und wurden zur Sicherung im Code gelassen, um im Falle eines Tenant Wechsels einfach auf eine andere Version wechseln zu können. Die letzte ID im roten Kasten muss der ID aus dem Browser entsprechen, sonst funktioniert der Datenaustausch nicht, da die App mit dem falschen Asset kommuniziert.

```

2
3 app.controller('TimeSeriesCtrl', ['$scope', '$http', '$interval', '$routeParams'
4   var self = this;
5   var scheduleId;
6   //self.assetId = "CA0C68033F744E9CB42142D0FAB6E081";
7   //self.assetId = "308B68BFF3F245EDA6B19C88D82A3629";
8   self.assetId = "6819A1EB7C8E406AB5AC30DAB1E4BE6D";
9
10
11 self.variableAliases = {
12   'FDI4TempCPU' : 'FDI4',
13   'FDI0TempCPU' : 'FDI0',
14   'FDI3TempCPU' : 'FDI3'
15 }
16
17 self.aspects = {
18   'Failsafe Temperatures' : {
19     'aspectName' : 'FailSafeModules',
20     'variables' : [
21       'FDI2TempCPU', 'FDI3TempCPU', 'FDI4TempCPU',
22     ],
23     'thresholds' : {
24       "thresholdYellow" : 55,
25       "thresholdRed" : 65
26     }
27   },
28   'Discrepancy' : {
29     'aspectName' : 'FailSafeModules',
30     'variables' : [
31       'FDI1Discrepancy',
32       'FDI1DiscrepancyOn',
33       'FDI1DiscrepancyType'
34     ],
35     'thresholds' : {
36       "thresholdYellow" : 50,
37       "thresholdRed" : 85
38     }
39   }
40 }

```

Abbildung 3.64: Timeseries

Im grünen Kasten müssen nun die Namen der Temperaturen so zugewiesen werden, wie sie im IoT Modeler stehen. Also zum Beispiel FDI4TempCPU gibt den genauen Namen für die App an. Diese greift dann wiederum über den IoT Modeler auf die definierten Datenbausteine in der SPS zu. Im gelben Kästchen sind die Werte für die Diskrepanz ersichtlich. Für alle Werte steht in den Kästchen der „aspectName“, dieser muss mit dem Namen aus dem gelben Kästchen in Abbildung 3.62 übereinstimmen.

4 Ergebnis

4.1 Bewertung von SPS und MindSphere

In diesem letzten Kapitel wird die Funktion von MindSphere analysiert und ein Ausblick mit Erweiterungsmöglichkeiten für die Zukunft gegeben. Dazu wird zuerst auf die Stärken und Schwächen der SPS-Anbindung an MindSphere eingegangen.

4.1.1 SPS in Aufbau & Inbetriebnahme:

Der Aufbau der SPS Hardware stellte kein Problem dar, das System von Siemens ist ein sehr anwenderfreundliches System, welches schon lange am Markt ist und daher sehr gut entwickelt ist. Die TIA Software zum Projektieren ist noch nicht so lange am Markt und läuft daher noch nicht immer einwandfrei. Es kann daher passieren, dass das Programm abstürzt oder zu langsam läuft. Dies kann aber auch an der in diesem Projekt verwendeten Entwicklerversion liegen. Durch regelmäßiges Zwischenspeichern kann dieses Problem minimiert werden. Die Software selbst ist sehr einfach zu erlernen und intuitiv zu bedienen. TIA bieten große Anzahl an vorgefertigten Bausteinen, welche grafisches Programmieren ermöglichen. Dadurch ist das Programm auch für nicht gelernte Programmierer oder für Personen, ohne Hochsprachenwissen einfach anzuwenden. Zusätzlich unterstützt die Live Schaltung auf die Steuerung den Programmierer und ermöglicht so eine einfache Fehlersuche.

4.1.2 SPS im laufenden Betrieb:

Befindet sich die Software einmal auf der SPS, gibt es im laufenden Betrieb keine Probleme mehr.

4.1.3 MindSphere in Aufbau & Inbetriebnahme:

MindSphere ist ein im Jahr 2017 auf den Markt gebrachtes Produkt. In diesem Projekt wurde die Version 2.0 verwendet. Der Aufbau und der Anschluss an das Netz sowie an die SPS sind einfach und ohne Probleme zu realisieren.

Probleme traten bei der Konfiguration des Gateways (MindConnect IOT 2040) auf. Dabei mussten die Konfigurationsdateien in MindSphere erstellt werden und per USB Stick auf die MindConnect gespielt werden. Leider kam es hier zu Komplikationen. Dabei traten vier Probleme auf.

1. Die Konfigurationsoberfläche funktioniert zwar, man muss aber wissen wo man hin klickt, da die Oberfläche oft Funktionen zeigt, die noch nicht funktionieren bzw. die Schaltflächen keinen Einfluss auf die Installation haben. Oft funktionieren Buttons nicht, da die Konfiguration zuvor gespeichert werden muss und in der Oberfläche falsch programmiert ist. Dass dieser Speicherbefehl erforderlich ist, ist nicht immer ersichtlich, was zu Störungen und zu Problemen bei der Konfiguration führen kann.
2. Eingabezeilen in der Oberfläche. Oft ist nicht ersichtlich ob nun Groß- oder Kleinbuchstaben eingegeben werden müssen. Da die Oberfläche alles in Kleinbuchstaben visualisiert, kann das zu Verwirrung führen.
3. Übertragen der Software auf das Gateway. Nachdem die Konfigurationsdateien für MindSphere erstellt wurden, können diese auf einen USB-Stick übertragen werden. Obwohl sich immer Dateien auf dem USB-Stick befinden, kann das Gateway diese nicht erkennen. Daher muss der Prozess oft wiederholt werden. Leider konnte auch mit Hilfe des Supports die Ursache des Problems nicht gefunden werden. Auch das Erstellen eines Programms zum Auslesen von Fehlern auf dem Gateway konnte keine Klarheit bringen. Sollte das Problem wieder auftreten, so ist die bisher einzig funktionierende Lösung den Prozess öfters zu wiederholen.
4. Problem bei der Einrichtung der Assets. Fügt man einen Datenübergabepunkt hinzu, funktioniert das ohne Probleme. Möchte man diesen jedoch löschen, bricht das System zusammen und alle Datenübergabepunkte müssen gelöscht und neu definiert werden. Das System speichert sich zwar die zuvor verwendeten Bezeichnungen, trotzdem sollte MindSphere in der Lage sein Datenpunkte zu löschen, ohne die restlichen Schnittstellen so zu beschädigen, dass diese ihre Funktion verlieren. Als Lösung und mit Rücksprache vom Support empfiehlt es sich, die falschen Datenübergabepunkte nicht aus dem System zu löschen. Treten Probleme auf, steht der Support immer hilfreich zu Verfügung.

4.1.4 MindSphere im laufenden Betrieb:

Auch im laufenden Betrieb funktioniert MindSphere noch nicht optimal. Abgesehen davon, dass die Oberfläche, mit den Browsern Firefox oder Internetexplorer sehr langsam sein kann, hat das System öfters noch Ausfälle beim Starten. Dabei werden die Daten nicht oder nur fehlerhaft übertragen. Die Gateways von Siemens haben zwar einen Pufferspeicher, trotzdem reicht dieser einige Male nicht aus oder wird nicht richtig befüllt. Läuft das System nach dem Einschalten, treten jedoch keine Probleme auf. Am besten wäre es daher, das System nach dem Einschalten dahingehend zu prüfen, ob die richtigen Daten übertragen werden. Im Falle eines Fehlers kann hier ein Restart helfen, am besten mit einer Trennung des Gateways von der Stromversorgung.

4.2 Erweiterungen

Demo-Case:

Der Demonstrator verwendet, wie oben beschrieben noch keine FSM, die analoge Signale verwerten können, um z.B. Temperaturdaten auslesen zu können. Siemens hat diese erst im Frühjahr 2018 vorgestellt und nicht für das Smart Factory Projekt freigeben. Der Demonstrator könnte zum gegebenen Zeitpunkt um diese Module erweitert werden. Einsetzen könnte man diese, indem man die Endlagenschalter durch Kraftmesssensoren ersetzt. Auch hier könnte wieder die Diskrepanz ermittelt werden, nur nicht die der Auslösezeit, sondern der Unterschied in den Kräften. Verwendet werden könnte hierzu wieder der UR5. Die App würde auch keine großen Anpassungen benötigen. Ein praktisches Beispiel wäre die Überwachung einer Zwei-Säulenpresse. Jeweils bei einer Säule befindet sich eine Kraftmessdose, die die sicherheitsrelevanten Daten an die SPS weitergibt. Ändern sich über den Verlauf der Zeit die Kräfte an den Sensoren kann zum Beispiel auf einen Verschleiß in den Führungen der Presse rückgeschlossen werden.

MindSphere:

Für MindSphere besteht die Möglichkeit, die neueste Version zu verwenden, die im Sommer 2018 auf den Markt kommt. Im Moment wird die Version 2.0 verwendet, welche auf der SAP Cloud läuft. Da diese teilweise sehr langsam ist und immer wieder Probleme auftreten, hat Siemens für Version 3.0 Amazon Web Services als Partner gefunden. Neben den oben angeführten Fehlern in der Eingabemaske, sollten mit Version 3.0 auch die Verbindungsprobleme behoben sein.

Der Wechsel zur Version 3.0 sollte einfach von statten gehen. Mit großer Wahrscheinlichkeit muss ein Firmware Update am Gateway durchgeführt werden. Zu hoffen ist, dass die Datenübergangspunkte in MindSphere übertragen werden können.

Falls diese Option nicht besteht, wäre es sinnvoll, ein Tool in die Version 3.0 zu integrieren, welches die konfigurierten Datenpunkte speichert, um in Zukunft den Aufwand bei einer Versionsänderung zu verringern. Im Fall dieser Arbeit lassen sich die Datenpunkte recht schnell neu eingeben. Betrachtet man aber eine große Industrieanlage mit hunderten von Sensoren kann diese Neukonfiguration viel Zeit in Anspruch nehmen.

4.3 Ausblick & kritische Betrachtung

In diesem Punkt sollen neben der Zusammenfassung noch ein paar kritische Gedanken angeführt werden.

Die Einrichtung von MindSphere auch für einen nicht IT Techniker gut möglich. Das Programmieren der SPS und die Bereitstellung der Datenpunkte erfordert Fachwissen im SPS Bereich.

MindSphere einfach mittels Plug und Play einzurichten und danach zu verwenden, ist auf keinen Fall möglich. Neben dem Fachwissen im SPS Bereich wird eine Einschulung mit dem MindSphere System benötigt, um fertige Apps aufzuspielen und die Datenpunkte dort

einzurichten zu können. Als Kunde könnte man diese Dienstleistung zukaufen, das System wird installiert und danach im Falle von Störungen von Experten gewartet. Generell funktioniert das System aber gut, bedenken sollte man, dass das Projekt noch in den Kinderschuhen steckt. Die Software Entwicklung ist ein kontinuierlicher Prozess, die Programme werden von Version zu Version besser.

Interessant als Anwender ist noch die Betrachtung der Langlebigkeit dieser Systeme. Wie lange so ein System betrieben werden kann, wie lange gibt es Updates, und wie lange steht die Hardware zur Verfügung?

Werden neue Systeme entwickelt, funktioniert die alte Datenübertragung ohne Probleme oder muss ein komplett neues System geschaffen werden? Ein weiterer interessanter Punkt wäre noch die Kosten/Nutzen Betrachtung für Industriebetriebe, ob sich die teure und aufwendige Einrichtung solcher Systeme für den Kunden überhaupt rentiert.

Auch gibt es für MindSphere starke Konkurrenz. Neben ähnlichen gesamtheitlichen Konzepten wie von General Electric (PREDIX) könnten auch die größeren Maschinenhersteller zu Mitbewerbern in diesen Anwendungen werden. Viele dieser Maschinenhersteller bieten schon ihre eigenen integrierten Systemlösungen im Bereich der Datenerfassung und -auswertung an. Eine vernünftige Weiterentwicklung von Siemens wäre daher, MindSphere Daten aus unterschiedlichen Cloud-Lösungen sammeln zu lassen. MindSphere sollte zusätzlich die einzelnen Anwendungen aller Hersteller bündeln können und so eine Gesamtlösung mit integrierten Fremdlösungen anbieten. MindSphere nennt sich zwar eine offene Plattform, dies bezieht sich jedoch nur auf die Möglichkeit, Applikationen auf MindSphere zu entwickeln und Datenpunkte mittels OPC-UA anzubinden.

Für die industrielle Anwendung müssten auch die im Moment zur Verfügung gestellten Apps deutlich ausgebaut werden. Zusätzlich wäre es sinnvoll, diese Siemens App so zur Verfügung stellen, dass z.B. Maschinenhersteller die angebotenen Apps weiterentwickeln können.

Nicht sinnvoll ist es, dass jedes Unternehmen seine eigenen Apps entwickelt, weil der Entwicklungsaufwand sehr groß sein kann.

Besser wäre es, eine Basis App mit Erweiterungsmodulen im Sinne eines Baukastens zur Verfügungen zu stellen, auf der individuelle Entwicklungen aufgebaut werden können.

Literaturverzeichnis

- [1] Günter Wellenreuther, Deiter Zastrow, Automatisieren mit SPS, Braunschweig: Viewegs, 2005.
- [2] S. AG, „Siemens Industry Support,“ 12 2014. [Online]. Available: https://cache.industry.siemens.com/dl/files/792/59191792/att_895924/v1/s71500_et200mp_system_manual_de-DE_de-DE.pdf. [Zugriff am 30 05 2018].
- [3] bild. [En ligne]. Available: <https://images.vogel.de/vogelonline/bdb/712900/712906/26.jpg>.
- [4] P. D. C. Siemers, „embedded-software-engineering,“ [Online]. Available: <https://www.embedded-software-engineering.de/echtzeit-grundlagen-von-echtzeitsystemen-a-669520/>.
- [5] *Functional safety of electrical/electronic/programmable electronic IEC 61508. 2010.*
- [6] J. Börcsök, Elektronische Sicherheitssysteme: Hardwarekonzepte,, Heidelberg, 2007.
- [7] L. S. Rappel, Simulation von Diagnosemaßnahmen von fehlersicheren Digitalausgängen einer SPS durch Einsatz eines virtuellen Prototypen, Hagenberg: Fachhochschule Hagenberg, 2015.
- [8] O. Asche, „Fourth-Generation Software,“ 1967.
- [9] DFKI. [Online]. Available: <https://www.datafox.de/details/datafox-und-industrie-4-0.de.html>. [Zugriff am 02 06 2018].
- [10] Volker P. Andelfinger, Till Hänisch, Industrie 4.0 Wie cyber-physische Systeme die Welt verändern, Annweiler: Springer, 2017.
- [11] [Online]. Available: <http://images.zeno.org/Meyers-1905/l/big/Wm04454i.jpg>. [Zugriff am 02 06 2018].
- [12] [Online]. Available: <https://opelpost.com/06/2014/fliessband-laubfrosch/#carousel-featured>. [Zugriff am 02 06 2018].
- [13] „Computer-automation,“ [Online]. Available: https://www.computer-automation.de/feldebene/antriebe/?gid=3480&_aid=93294&cp=3. [Zugriff am 02 06 2018].
- [14] „Industrie.de,“ [Online]. Available: <https://industrie.de/top/6637/>. [Zugriff am 02 06 2018].
- [15] D. C. d. Ideenwerkstatt, Industrie 4.0 Controlling im Zeitalter der intelligenten Vernetzung, 2015.
- [16] Kagermann, Umsetzungsempfehlungen für das Zukunftsprojekt, 2013.
- [17] D. A. d. Technikwissenschaften, „Umsetzungsempfehlungen für das ZukunftsprojektIndustrie 4.0,“ 2013.

-
- [18] Timothy Grance, Wayne Jansen, „Guidelines on Security and Privacy in Public Cloud Computing,“ *National Institute of Standards and Technology*, pp. 800-144, 2011.
- [19] B. - . B. f. S. i. d. Informationstechnik, „Cloud Computing Grundlagen,“ [Online]. Available: https://www.bsi.bund.de/DE/Themen/DigitaleGesellschaft/CloudComputing/Grundlagen/Grundlagen_node.html. [Zugriff am 02 06 2018].
- [20] „Sevone,“ [Online]. Available: https://www.sevone.com/sites/default/files/images/cloud_services_model_graph.jpg. [Zugriff am 02 06 2018].
- [21] „Wirtschaftslexikon,“ [Online]. Available: <https://wirtschaftslexikon.gabler.de/definition/internet-der-dinge-53187/version-276282>. [Zugriff am 09 2018].
- [22] D. Bundestag, „Aktueller Begriff Industrie 4.0,“ Berlin, 2016.
- [23] D. F. S. AG, „MindSphere - the cloud-based, open IoT operating system. Your way to digital services“, 2017.
- [24] A. Gilchrist, *Industry 4.0: The Industrial Internet of Things*, Apress, 2016.
- [25] „Computerwoche,“ [Online]. Available: <https://www.computerwoche.de/a/was-ist-was-bei-predictive-analytics,3098583,4>. [Zugriff am 12 06 2018].
- [26] E. Egger, *Cloud Applikationen in Industrie 4.0*, Hagenberg, 2017.
- [27] „Siemens Mindsphere Homepage,“ [Online]. Available: <https://www.siemens.com/global/de/home/produkte/software/mindsphere.html>.
- [28] „Siemens industry support,“ [Online]. Available: <https://support.industry.siemens.com>.
- [29] R. E. Silverman, „The words of Tomorrow,“ 2000.
- [30] U. Robots, „Benutzerhandbuch UR5/CB3,“ Universal Robots.
- [31] M. a. peshkin, „Cobots: Robots for collaborataion with people,“ 2016.
- [32] [Online]. Available: <https://www.spsshaus.ch/Schulungen-Kursangebot-TIA-SCL2.html>. [Zugriff am 04 06 2018].
- [33] „SPS-Lehrgang,“ [Online]. Available: <https://www.sps-lehrgang.de/und-oder-verkneuepfungen-in-fup/>. [Zugriff am 04 06 2018].
- [34] S. AG, „MindSphere – Getting connected to MindSphere,“ Digital Factory, NÜRNBERG, 2017.
- [35] „ZID,“ [Online]. Available: <https://zid.univie.ac.at/dhcp/>. [Zugriff am 04 06 2018].
- [36] „bigdata-insider,“ Was ist OPC-UA, [Online]. Available: <https://www.bigdata-insider.de/was-ist-opc-ua-a-698144/>. [Zugriff am 04 06 2018].
- [37] Mozilla, „<https://developer.mozilla.org/de/docs/Web/Manifest>,“ 07 2018. [Online].
- [38] A. Gilchrist, *The Industrial Internet of Things*, 2016.

Abbildungsverzeichnis

Abbildung 2.1: Speicherprogrammierbare Steuerung [2].....	2
Abbildung 2.2: SPS dezentraler Peripherie [3].....	3
Abbildung 2.3: Ablauf der zyklischen Bearbeitung eines SPS Programms [1].....	5
Abbildung 2.4: 1oo2 Schaltprinzip [6]	6
Abbildung 2.5: Blockschaltbild einer fehlersicheren Baugruppe [7].....	6
Abbildung 2.6: Platine einer F-DQ Baugruppe	7
Abbildung 2.7: 4 Stufen der industriellen Revolution [9]	8
Abbildung 2.8: Dampfmaschine [11].....	9
Abbildung 2.9: Die erste Fließbandproduktion von Opel [12].....	9
Abbildung 2.10: Erste Simatik SPS von Siemens [13]	10
Abbildung 2.11: Kerntechnologien für die Industrie 4.0 [14]	10
Abbildung 2.12: Merkmale der Industrie 4.0 [15].....	11
Abbildung 2.13: Wertschöpfungsnetzwerk [16]	11
Abbildung 2.14: Durchgängiges System-Engineering [16].....	12
Abbildung 2.15: Vertikale Integration [16].....	12
Abbildung 2.16: Darstellung der verschiedenen Servicestrukturen [20].....	16
Abbildung 2.17: Wachstumsverlauf der Assets bis 2020 [23].....	17
Abbildung 2.18: Vergleich von M2M und IIoT [26]	18
Abbildung 2.19: Symbolische Darstellung von Siemens MindSphere [23].....	19
Abbildung 2.20: Drei-Schichten-Modell von MindSphere [23].....	20
Abbildung 2.21: MindConnect IoT 2040 [28]	21
Abbildung 2.22: MindConnect Nano [28].....	21
Abbildung 2.23 : Datentransport	23
Abbildung 3.1: Gesamtaufbau	25
Abbildung 3.2: UR5 [30]	26
Abbildung 3.3 On Robot Greifer.....	26
Abbildung 3.4 Not-Aus-Taster.....	28
Abbildung 3.5 Demo-Case Diskrepanz.....	30
Abbildung 3.6: Datenfluss der Diskrepanz Sensorik	31
Abbildung 3.7 Demo-Case Temperaturüberwachung.....	32

Abbildung 3.8 : Zentrale Peripherie	33
Abbildung 3.9 : Dezentrale Peripherie	34
Abbildung 3.10: Aufbau einer Moduleinheit	36
Abbildung 3.11: SPS-Hardware Unterbau	37
Abbildung 3.12: MindConnect 2040.....	37
Abbildung 3.13: Gegenüberstellung von FUP und SCL [32] [33].....	38
Abbildung 3.14: Netzwerkplan und Grundaufbau der SPS	39
Abbildung 3.15: Dezentrale Peripherie MindSphere 2	39
Abbildung 3.16: Dezentrale Peripherie MindSphere 1	40
Abbildung 3.17: Zuweisung von Zugängen	41
Abbildung 3.18: Channel Settings.....	43
Abbildung 3.19: Einstellungen Diskrepanz-Case	44
Abbildung 3.20: Zyklische Durchlauf der Bausteine [33].....	45
Abbildung 3.21: Im Programm verwendete Bausteine	46
Abbildung 3.22: OB 1	47
Abbildung 3.23: FB_ControlCabinet_HeatingTimer	48
Abbildung 3.24: FB_EmergencyStopButton_OperationCounter	49
Abbildung 3.25: FB_EmergencyStopButton_UsedInterval	50
Abbildung 3.26: FB_showDiscrepancy	51
Abbildung 3.27: Main_Safety_RTG1	52
Abbildung 3.28: Datenbaustein ohne MindSphere Anbindung	52
Abbildung 3.29: Datenbaustein mit MindSphere Anbindung.....	53
Abbildung 3.30: Hardware Nummer	54
Abbildung 3.31: MindSphere Data Control	55
Abbildung 3.32 Datensatz	56
Abbildung 3.33 DB101 MindSphere Word.....	56
Abbildung 3.34: IoT 2040 [34].....	58
Abbildung 3.35: Stromanschluss [34]	58
Abbildung 3.36: MindSphere Startseite mit Standard Apps (rote Kästchen)	59
Abbildung 3.37: Data Modeler	59
Abbildung 3.38: Dateneingabefeld.....	60
Abbildung 3.39: Hardwareauswahl	60

Abbildung 3.40: Hardware ID Eingabe	61
Abbildung 3.41: Schnittstellenkonfiguration Fabrik seitig	61
Abbildung 3.42: Schnittstellenkonfiguration zur Cloud	61
Abbildung 3.43: Online Status	62
Abbildung 3.44: Aspects Konfiguration	63
Abbildung 3.45: MindSphere Datenbaustein	64
Abbildung 3.46: Startseite von MindSphere	65
Abbildung 3.47: Temperaturanzeige.....	66
Abbildung 3.48: Auswahl der gewünschten Daten.....	66
Abbildung 3.49: Screenshot der Konsole, Login und Passwort (passcode).....	68
Abbildung 3.50: Passwortgenerierung im Browser	68
Abbildung 3.51: Zwei-Faktor-Authentifizierung.....	69
Abbildung 3.52: Screenshot der Handy App	69
Abbildung 3.53: Authentication Code	69
Abbildung 3.54: Screenshot der Konsole	70
Abbildung 3.55: Eingabe des Pfades.....	70
Abbildung 3.56: App pushen	71
Abbildung 3.57: Introduction Application Manager.....	71
Abbildung 3.58: Eingabe der Daten im Application Manager.....	72
Abbildung 3.59: Healthmonitor App	73
Abbildung 3.60: Discrepancy Messung	73
Abbildung 3.61 Manifest.....	74
Abbildung 3.62: IoT Modeler Datenpunkte und Namen	75
Abbildung 3.63: Asset ID.....	75
Abbildung 3.64: Timeseries.....	76

Tabellenverzeichnis

Tabelle 1: Tabelle der verwendeten Komponenten.....	35
Tabelle 2: Datentypen	65

Abkürzungsverzeichnis

A	Amper
AI	Analogeingabebaugruppe
AQ	Analogausgabebaugruppe
CPU	Central Processing Unit
DB	Datenbaustein
DI	Digitaleingabebaugruppe
DHCP	Dynamic Host Configuration Protocol
DQ	Digitalausgabebaugruppe
FB	Funktionsbaustein
FC	Funktion
FSM	Fail-Safe Modul
FUP	Funktionsplan
HTML5	Hypertext Markup Language
IaaS	Infrastructure as a Service
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP Adresse	Internet Protokoll Adresse
LAN	Local Area Network
M2M	Machine to Machine
Mac-Adresse	Media-Access-Control-Adresse
NIST	National Institute of Standards and Technology
OB	Organisationsbaustein
OPC-UA	Open Platform Communications Unified Architecture
PaaS	Platform as a Service
PLM	Product Lifecycle Management
PROFINET	Process Field Network
S7	STEP 7
SaaS	Software as a Service
SCL	Structured Control Language
SD-Karte	Secure digital memory Card
SIL	Safety Integrity Level
SPS	Speicherprogrammierbare Steuerung
TIA	Totally Integrated Automation
V	Volt

Anhang

Anhang 1: Datensatz 241 F-AI 89

Anhang 2: Datensatz 240 F-DI 90

